Modelling and optimisation of pharmaceutical formulations and processes

Zolotariov, Eyal

Modelling and Optimisation of Pharmaceutical Formulations and Processes

A thesis submitted by
Eyal Zolotariov

**to the University of London
for the degree of Doctor of Philosophy
in the Faculty of Medicine**

Department of Pharmacy
King's College London
April 2001

To my family

Acknowledgments

My gratitude to the thesis advisor Dr. Jamshed Anwar, especially for the freedom he gave me in conducting this research.

Many thanks to the friends in the Computational Pharmaceutical Sciences group, especially to Dr. Papa Boateng for the support he gave me.

Thanks to Dr. D. Barlow, Dr. C. Richardson & Mr. V. Dawes from the pharmacy department. From other departments at King's College London, I want to thank two members of staff, Dr. M. Plumbley & Dr. P. Milligan, each one helped me in his field of expertise.

To my wife, for standing beside me patiently throught the duration of this study, for supporting and giving me practical help regarding this research.

I want to thank my parents for the moral support they gave me. Special thanks to my father for his ideas regarding this study and for my family's financial support, without which I could not have done this study.

Abstract

The pharmaceutical formulator has a problem of setting up many variables that influence conflicting dosage form properties. The formulator should aim to get towards the optimum properties of pharmaceuticals. One should arrive at this optimum point in a minimum number of experiments to save time and money. To address this problem mathematical optimisation techniques and expert systems have been utilised. In general, work in the area of mathematical optimisation techniques has focused on primarily the regression model to address this problem. Recent work has suggested the ANN as a better model for optimisation of formulation variables. The comparison between ANN and regression has traditionally favoured the former model (e.g. Hussain et al., 1991). This work examines the statistical basis of these earlier works and advances upon them.

The use of ANN and regression to model tablet properties was examined in the first study. Several problems were examined. It is apparent that the use of different validation methods will give different results. It was not possible to reduce the number of validation experiments by training ANN on all the data, measuring error and deciding from that value the best ANN topology. This study also suggests that scaling the data is critical for effective learning in ANN. The same data used in the first study was used in the following one. For improving ANN predictive ability different training methods were used apart from simple backpropagation. It was shown it is worth using different training methods since the predictive ability improves. After this comprehensive work was done, the best ANN models were compared with the predictive ability of the best regression models of the first study. This comparison was extensive and various aspects were examined like, if ANN predictive ability is better than regression, is this statistically significant? Which method predicts better the extreme values of the responses? Is ANN predictive ability more susceptible to the response value than regression? Does the predictive ability of ANN improve if modelling each response one at a time—thus reducing ANN complexity—instead of all responses simultaneously? The answers to these questions are complex and are addressed in the thesis. The same questions were tackled in the next study on limited data. As opposed to the tablet properties study, this study on capsules dealt with experiments that were not well designed, so the data was limited. Indeed, ANN and regression models succeeded in predicting only 5 out of the 9 response variables. This study showed that the answer to the question - is it worth modelling an ANN one response at a time depends on the system the ANN tries to model. In these 3 studies there was

different emphasis on the ANN variables manipulated.

The following study took the raw data from the tablets study. Multiobjective optimisation was done after ANN was trained. The optimisation was for two conflicting responses of disintegration time and friability. The computer generated a solution that tried to satisfy the two responses. This solution was achieved using an optimisation method that incorporates ANN with the Goal Attainment method. This new multiobjective optimisation algorithm runs successfully on the computer. In order to solve tablet formulation problems an expert system was developed. It demonstrates how ANN/Regression models are used through the process of data collection and decision making in the field of tablet formulation. The expert system that was generated in this study, apart from modelling and optimisation tools, also incorporates database and heuristic rules in one application called Expha expert system.

Do ANN predict better than regression? The answer to this depends on the aspect of comparison studied. It is not possible to state categorically whether ANN or regression is better.

Table of Contents

List of Tables

List of Figures

List of Abbreviations / Symbols

A_ - Artificial Neural Networks (this term is not a stand alone but attached to response)

ABS – Absolute Value

ANN – Artificial Neural Networks

AUC - Area Under the Dissolution Curve

B – Bias

BFG - BFGS quasi-Newton method

BFGS - The Quasi-Newton method of Broyden. Fletcher, Goldfarb and Shanno

BGD - Basic Gradient Descent

BR - Bayesian Regularization

Carr - Carr's Compressibility Index

CFV - Coefficient of Fill Weight Variation

CGB - Powell-Beale Conjugate Gradient Algorithm.

CGF - Fletcher-Reeves Conjugate Gradient Algorithm

CGP - Polak-Ribiere Conjugate Gradient Algorithm

d^2 - interranging communalities

dc - Drug Concentration

CP - Carboxyvinyl Polymer

DC – Direct Compression

Disint – Disintegration Time

Dissol – Dissolution Rate

D (ps) - Drug Particle Size

D (sol.) - Drug Solubility

DIF_ - The Difference between ANN and Regression Absolute Percent Deviations from Observed Values (this term is not a stand alone but attached to response)

dl - Disintegrant Level

dt - Disintegrant Type

DT - Disintegration Time

Er_Fri – Erosion Friability

ft - Filler Type

fl - Filler Level

gl - Glidant Level

GDM - Gradient Descent with Momentum

GDX - Gradient Descent with Adaptive Learning Rate

GFF-MLP - Generalised Feed Forward Multi-Layer Perceptron Network

GSH - Galenical Development System Heidelberg

GUI - Graphical User Interface

H - Hausner's ratio

Hardn – Hardness

HPE - Handbook of Pharmaceutical Excipients

Im_Fri – Impact Friability

IPC – In-Process Control

k - Decomposition Constant

ll - Lubricant Level

OCR - Optical Character Recognition

OSS - One Step Secant Method.

MDT - Mean Dissolution Time

MLP - Multi-Layer Perceptron Network

PE – Perceptron

LM - Levenberg-Marquardt algorithm

MSE - Mean Squared Error

MRE - Mean Relative Error

NDA - New Drug Application

p-value - The Observed Significance Level

PDFT - Pharmaceutical Dosage Forms: Tablets

PRCG - Polak-Ribiere conjugate gradient algorithm

PVP – Polyvinylpyrrolidone

QA - Quality Assurance

QC - Quality Control

QSAR - Structure Activity Relationship

R_ - Regression (this term is not a stand alone but attached to response)

r^2 - The Coefficient of Determination

R^2 - The Multiple Coefficient of Determination

RB - Radial Basis

RBF - Radial Basis Function

RD - Residual Difference

Reg - Regression

RMS - Root Mean Squared Error

RP - Resilient Backpropagation

RSM - Response Surface Methodology

SC - Spread Constant

SCG - Scaled Conjugate Gradient Algorithm

SOFM-MLP - Self-Organising Feature Map Multi-Layer Perceptron Network

SS - Sum Squares

SSE - Sum Squared Error

SSW - Sum Squared Weights

Tensil - Tensile Strength

Thickn – Thickness

VDT - Variance of the Dissolution Time

Vmax - Maximum Bulk Density

Vmin - Minimum Bulk Density

W - Weight

Weight - Mean Weight

X - Independent Variable(s)

Y - Dependent Variable(s)

°C - Degrees in Celsius

Λ - Objective Function Region

Ω - Feasible Region

$\Re^3$ - Three Dimensional Parameter Space

1. Introduction

1.1 Statement of the problem

Formulation development is a problem that has been of concern to pharmaceutical researchers since the early days of pharmaceutical industry. The problem is now even more challenging since the regulatory authorities enforce that the pharmaceutical product complies with the limits of the pharmacopoeia. The pharmaceutical formulator has a problem of coping with the many variables that influence dosage form properties often in a conflicting manner. For example, a common problem is that a tablet has to disintegrate quickly and be fast dissolving yet must be hard enough to remain intact before ingestion. If the formulator considers just the hardness the compaction force is increased and the tablet made harder but it may not be fast dissolving. The formulator should aim to get the optimum properties of a pharmaceutical formulation. The formulator should arrive at this optimum point in a minimum number of experiments to save time and money. He/she has to evolve towards the tablet formulation solution with a path that maintains the knowledge acquired. This will prevent the situation where there is no solution on the horizon after conducting many experiments. This removal of uncertainty regarding experiments is critical when submitting a research proposal to the management. Another issue is that there is a loss of expertise due to formulators moving from one place to another or retirement. This raises the problem of how to retain pharmaceutical formulation knowledge.

To address these problems mathematical optimisation techniques and expert systems have been developed. However, in pharmaceuticals the use of these techniques is relatively new and there are many questions that still remain to be resolved. Which of the modelling techniques regression or Artificial Neural Networks (ANN) should be used? How should one validate the model that describes the pharmaceutical problem? What criterion to employ in identifying the superior model? How should one decide that the chosen model is

adequate? Is an ANN better than regression for modelling of pharmaceutical formulation problems? How should one compare these two modelling techniques? What is the best optimisation method? Are expert systems effective or likely to be effective in pharmaceutical development and how do ANN and regression models relate to them? Some of these aspects have been discussed in previous works (all of them were addressed in this study). There were however important questions still to be addressed like how to compare ANN and regression, or how to decide whether a model has genuine predictive ability. In addition, this study examined critically studies related to the problems of solving the questions presented earlier. This study proposes methodology for choosing the best model that describes the data. The method developed was implemented for the tablet formulation process in this work by developing Expha, an expert system. Another goal was to incorporate the two domains of pharmaceutical knowledge and data modelling into one coherent application.

In this introduction the need for modelling, optimisation and expert systems will be discussed. Afterwards, there will be a quick review of regression and ANN methodologies. The similarities and differences between these methodologies will be examined. Then the development of expert systems and associated literature will be reviewed. Next, the applications of these techniques in pharmaceutical development will be reviewed. Then, issues brought about by differences between this study and others, and the contribution of this study will be discussed. The final part will describe the structure of the thesis.

1.2 The need for modelling, optimisation and expert systems

The pharmaceutical formulator whether aware of it or not is using three basic elements: Heuristic rules, a database and an optimisation based on data modelling. A typical formulator may use their memory to remember many excipients and their advantages and limitations. They may also use articles and computerised databases. The use of heuristic rules gathered and developed through expertise are called rules of thumb. Usually, the rules are remembered and the formulator is not always aware that these serve as a guide when a new formulation is invented. These two steps are invoked automatically by the formulator each time a new formulation is prepared. The prepared formulation is tested and response variables like time to dissolve and hardness are measured. If the result does not satisfy the

requirements, a change in quantities of excipients or process variables with a trial and error approach is adopted. The formulator might also build up a model by gathering the information from experiments designed to help build the model. The model that is built will give suggestions for a better formulation. The latter part belongs to the optimisation part.

Upon trying to simplify the problem further there are two distinct domains in the formulation problem. They are the pharmaceutical knowledge base of the problem that is the database part, e.g. interactions of the drug with certain excipients, and the heuristic rules. The second part is the modelling which is statistical in its nature. The problem with the first part is that the formulator may either not be aware of certain aspects that are documented in the literature, or might forget to look at the recommended concentration of the excipient and/or its interactions with the drug. The formulator might stick to a certain pattern of making the formulation without considering a new method of formulation with a completely different range of excipients. It is quite a common problem that people want to be on the safe side which not only produces inferior formulations relative to what could be achieved, but also presents a problem that can only be solved with considerable difficulty with their limited knowledge. On the statistical part of building up a model they might be afraid to touch the subject and prefer to avoid it. Another possibility is that they may misuse the statistical tool of data modelling. Even the most qualified formulator who keeps up to date on new materials has a problem of remembering and implementing all their knowledge about formulation. Remembering could mean just remembering to look in the correct book for the relevant information; or to remember some previously-read important information regarding a certain excipient that is being used in the present formulation.

The problem is that to achieve a better formulation the formulator has to spend more time in investigating the formulation problem. In general, formulators lack methodologies or tools that will allow them to save time, or to enable them to solve formulation problems that they did not succeed in solving before. Saving time means not simply because with the new approach the new formulation is developed faster but because the better formulation that is generated may stay a longer time on the market until there is a need to change it to new requirements. A common example is disintegration of tablets. As there is a relationship between disintegration rate and dissolution rate, tablets were in the past optimised to achieve fast disintegration and so were able to achieve the desired dissolution. The tablet formulations optimised in this manner may be permissible according to a new

pharmacopoeia without any changes in their formulation.

1.3 Regression analysis

ANN and regression analysis are both used in analysing results of pharmaceutical experimentation. The ability of ANN and regression analysis to characterise experimental relationships makes them useful in the process of optimisation of pharmaceutical formulations. Both of them are used to describe functions of the response variables. The response variables are often termed the dependent variables (Ys). The arguments of these functions are the independent variables (Xs). Reverting to the example in Section 1.1, disintegration time and hardness are dependent variables, whereas the amounts of disintegrant and lubricant are the independent variables. The researcher can control the independent variables and by varying them can influence the response or dependent variables.

Regression will be discussed here very briefly. The reader who wants comprehensive foundation to this subject should consult Draper & Smith's book (Draper & Smith, 1987). The reader who wants more specific information to the current domain about regression and design of experiments could find help in Armstrong & James book (Armstrong & James, 1996). Regression is a statistical technique and is a process of finding a mathematical model (an equation) that best fits the data (Mendenhall & Sincich, 1996). The regression equation is fitted to the data according to the least squares method. The least squares line is one that satisfies the property:

That the sum of squares error (SSE), $\Sigma(Y_{predicted} - Y_{observed})^2$, is smaller than any other line. In more simple terms, if one wants to draw a linear (straight) regression line manually he will draw a line that seems to him the closest one to the data points. The mathematical description of the line is the regression equation. Surprising as it may seem, beginning the analysis of data by drawing a line manually can prevent errors. The human eye captures the overall trend of the data ignoring outliers. Whereas if someone does this simple line with a computer, they will often get a line that does not represent the data trend, since the computer takes into account all data points if not programmed specifically to do otherwise.

Regression is used in many fields. From the various fields of life sciences to areas that are considered less scientific like economics. In economics it may be used to predict how

investment in advertisement will influence sales. Sometimes this tool is used in a biased way and subjective conclusions are derived that can often be read in the daily newspapers. For example, a woman getting married when pregnant will probably suffer financial problems. In this case only one parameter was measured (x) whether the woman was pregnant when she was married and the response (y) that was measured was the woman's economic state after a few years. The layman may think that if his/her daughter gets married when she is pregnant probably she will suffer from poverty. However, he should not jump to the latter conclusion, since a correlation does not imply a causal relationship.

The order of a regression model is an important issue necessary to understand which models were built in this study. This approach presented here to model order is quite common (Mendenhall & Sincich, 1996; Biles & Swain 1980; Box et al., 1978). A first order model is a straight line model with the following equation as an example:

$y = a_0 + a_1*x_1 + a_2*x_2 + a_3*x_3$.

The three x_j variables stand for the independent variables being manipulated, a_0 the equation intercept and the other a_i values are the regression parameters and these four values calculated by the least-square method. A second order model of three independent variables will look like:

$y = a_0 + a_1*x_1 + a_2*x_2 + a_3*x_3 + a_4*x_1*x_2 + a_5*x_1*x_3 + a_6*x_2*x_3 + a_7*x_1^2 + a_8*x_2^2 + a_9*x_3^2$.

Notice that the interaction term $x_1 * x_2 * x_3$ is not included since it is considered as third order term. To summarise, the order of model is according to the maximum power of the independent variables, or the maximum number of independent variables multiplied with each other in the regression equation.

Variable selection is a method of controlling the way in which variables are entered into and removed from, the regression equation. The criteria for selection of variables is the probability of F denoted by the letter p (F is a type of distribution). The p-value is used extensively in statistical tests (used frequently in this study) and is also called 'the observed significance level'. When a statistical test is conducted the p-value is the probability of observing a value of the test statistic at least as contradictory to the null hypothesis as the observed value of the test statistic, *assuming the null hypothesis is true* (Mendenhall & Sincich, 1996). In other words, when the p-value is 0.05 then there are about 5 chances in 100 that we would reject the null hypothesis although it is the true hypothesis. We are about 95% confident that we have made the right decision. In regression, the null hypothesis

which is tested regarding each coefficient is that its value is zero and as such is not contributing to the prediction of the response variable, i.e. if the p-value of regression coefficient is less than the confidence level decided (typical value is 0.05), then the null hypothesis can be rejected.

Three different methods can be used for building the regression equation:

1. Choose the order of the regression equation without any selection of variables.
2. Stepwise regression: selection of variables proceeds by steps. At each step, variables already in the equation are evaluated according to the selection criteria for removal; also variables not in the equation are evaluated for entry. This process is repeated until no variable is eligible for entry or removal.
3. Backward elimination: all variables that are in the equation are evaluated according to the selection criteria for removal. Those eligible are removed one at a time until no more are eligible (Norusis, 1997)

Generally, stepwise regression is considered to be the best way of variable selection. It is better than backward elimination because it is possible to consider starting with a very big model in which the number of coefficients is bigger than the number of data points. Whereas in backward elimination one is limited by the fact that the number of coefficients in the starting model cannot exceed the number of data points. Hence, the number of models that the stepwise regression model checks in the variable selection procedure can be much higher. Nevertheless, in reality one does not know from the beginning which method of variable selection is appropriate. And the reason for this is the fact that the disadvantage of a method like stepwise regression is the big number of statistical tests that are conducted in each of which, it is possible that an error may be made.

1.4 ANN

ANN are networks of adaptable nodes which through the process of learning from task examples store experimental knowledge and make it available to use (Aleksander & Morton, 1995). The knowledge is stored by the weights in the ANN and is made available to use by the possibility of feeding the trained ANN with certain inputs and the network would give appropriate outputs.

1.4.1 Uses of ANN

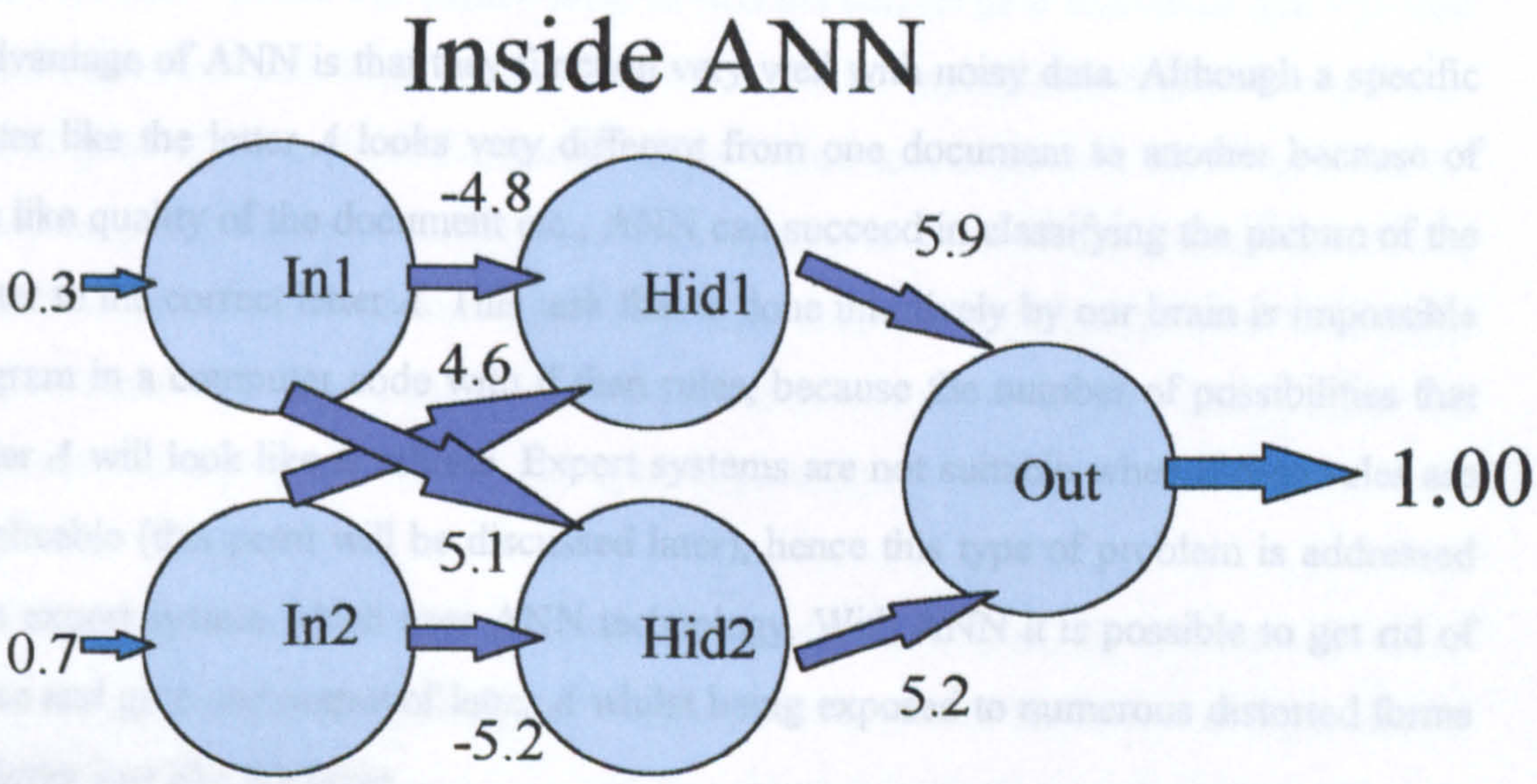
The use of ANN is widespread today in many areas. Banks use them to identify customer's signatures on cheques to avoid faked signatures. The computer does this job better than human experts. This example demonstrates why ANN systems are often called expert systems – because they replace human experts. ANN are very important tools for classification problems. In optical character recognition (OCR) software the computer captures the picture and it has to classify each letter it scans to match the letter in the alphabet it most resembles. After the ANN classification of the letters, it can be viewed and edited in a word processor. The less sharp the picture given to the computer, the more the software becomes prone to errors such as replacing the letter o with the number 0 (zero), and this is more challenging to the software since the classification problem has become more difficult. Even the simplest OCR software upon scanning a document printed on laser paper rarely has errors. A related problem is the task of voice recognition. The similarity to the OCR problem is obvious. In both of them the computer interprets data related to our senses, vision for OCR and hearing for voice recognition. In a voice recognition problem by receiving our voice the computer turns our voice into characters with voice recognition software. The recognition of a human face is a classification problem of pattern recognition just like OCR. For solving that problem, ANN that recognises a human face was built (Aleksander & Burnett, 1987). The ANN was trained on different facial expressions of the same man and was also able to identify the man with facial expressions it was not trained on, e.g. recognise the man by his funny face although in the training phase he never smiled.

Since ANN are so good in solving classification and modelling type problems, they have been used in many fields like engineering, science, mathematics, medicine, business and literature. This section will describe the uses of ANN in fields related to medicine and especially to pharmacy. In cancer diagnostic uses, ANN have been used to predict the recurrence of breast-cancer (Ravdin et al., 1992). ANN are in use for the screening of new drug applications (NDA) candidates for anti-cancer drugs (Weinstein et al., 1992). In cardiac diseases, ANN can predict the occurrence of acute myocardial infarction according to cardiac enzymes (Furlong et al., 1991). ANN are of use in the field of structure activity relationship (QSAR) (Tetko et al., 1993) which deals with the correlation of biological activity to various physiochemical parameters. ANN are employed in the field of molecular graphics (Livingstone et al., 1991). The interpretation of analytical data could be done using

ANN in various analytical fields like HPLC (Metting & Coenegracht 1996) or UV (Bohm et al., 1992). ANN are in use for the prediction of protein structure from its amino acid sequence (Holley & Harplus, 1991). The uses of ANN in the fields of biopharmaceutics (e.g. in vitro-in vivo correlation), pharmacokinetic, pharmacodynamic and pharmaceutical formulation will be discussed further on in this chapter.

1.4.2 Inside ANN

To give a greater understanding of how ANN function inside, an example of an ANN is given in Figure 1.1 which illustrates in detail how information is processed in a typical ANN.



Equation 1 $$f(x) = 1 / (1+e^{-x})$$

Calculation in "Hid1" $0.3 * (-4.8) + 0.7 * 4.6 = 1.78$ $f(x) = 0.86$

Calculation in "Hid2" $0.3 * 5.1 + 0.7 * (-5.2) = -2.11$ $f(x) = 0.11$

Calculation in "Out" $0.86 * 5.9 + 0.11 * 5.2 = 5.646$ $f(x) = 1.00$

Figure 1.1: Inside ANN

The ANN is fed with the inputs 0.3 and 0.7; 0.3 goes into input neuron 1 ("In1") and 0.7 goes into input neuron 2 ("In2"). Hidden neuron 1 ("Hid1") receives its inputs from the two input neurons but these values are first multiplied by the 'synaptic strength' (weight) between each of the input neurons and hidden neuron 1. The weighted inputs are then summed up in the hidden neuron 1 (Calculation in "Hid1"). The result is entered into the activation function (Equation 1) and the output of hidden neuron 1 is 0.86. A similar calculation is carried out in hidden neuron 2 (Calculation in "Hid2"). The outputs from "Hid1" and "Hid2" are multiplied by the corresponding weights and serves as input to the output neuron. The previous calculation is repeated in this neuron (Calculation in "Out") to yield the final output. The learning process is done by comparing the output of the ANN to

the desired output and adjusting the weights accordingly. This process is repeated until the ANN has learned. Backpropagation is commonly used to adjust the weights. The weights adjustments are propagated backwards from the output neurons to the input neurons.

1.4.3 ANN in relation to brain and computers

The advantage of ANN is that they function very well with noisy data. Although a specific character like the letter *A* looks very different from one document to another because of factors like quality of the document etc., ANN can succeed in classifying the picture of the character to the correct letter *A*. This task that is done intuitively by our brain is impossible to program in a computer code with if then rules, because the number of possibilities that the letter *A* will look like is infinite. Expert systems are not suitable when if-then rules are not applicable (this point will be discussed later), hence this type of problem is addressed with an expert system which uses ANN technology. With ANN it is possible to get rid of the noise and give one output of letter *A* whilst being exposed to numerous distorted forms of this letter just like the brain.

The method by which the brain learns is feedback. When a child does something that is seen as positive he/she is given positive feedback. When something is seen as wrong he/she is given negative feedback. The classic example is of Pavlov who trained a dog for conditioning behaviour. Each time he rang a bell the dog was given food. After several times each time the dog heard the bell it was expecting the food as shown by the saliva from its mouth. So it is possible to train the brain to respond to specific stimuli in a specific manner. A dog can be trained to produce saliva each time it hears a bell and a child can be trained to behave in the desired way. In the same manner it is possible to train a computer that will respond in a desired manner to specific stimuli. The desired manner is the output and the stimuli is the input. Just like the dog and the child the computer needs training examples which will guide it to the correct output with a certain input.

One can perceive ANN and database systems also as devices for storing information. ANN can be trained to memorise the data and every time the ANN is asked by the user by putting certain input to the system it will give output with error close to 0. In the same way data may be stored in the memory of the computer and every time pull out specific data by addressing the correct address in the computer memory. There are two basic differences

between the location of the data in ANN and in computer system. In computer memory there is a physical location for every item in memory, in ANN just like the brain there is no specific place for every memory item and the memory is dispersed.

Karl Lashley (1950) conducted experiments to find out the location of memory in various animals, especially in rats. He trained them to find the correct path in a maze that will lead them to the food reward. After training about 100 rats he began removing various parts of the cortex (part of the brain). Without any significance to the location of the part that he removed, the performance of the rats did not change up to a maximum removal of 10% of the cortex. When he removed more than 10% their mistakes were proportional to the size of the part removed but not to its location. Lashley was astonished to find out that the memory stored in the cortex was not in a specific place but everywhere in the cortex.

Trained workers, with parts of their brain destroyed in accidents, do not lose all brain-functioning capability. Documentation of people that lose part of their brain and yet did not stop functioning completely exists. More serious damage leads to a gradual degradation in the brain activity without it abruptly stopping from functioning; but the location of the damaged area is also an important factor of this pattern. This phenomena is similar to what happens in ANN as the network loses more and more neuron units (also termed perceptrons and abbreviated as-PE's) (Rumelhart & McClelland, 1986a).

Rumelhart & McClelland (1986b) trained a network to output the past tense of verbs in English. The ANN was able not just to add the common 'ed' automatically. After the network was trained it was examined on verbs which it had not seen before, like 'guard' and 'weep' and it succeeded of determining their past tense as 'guarded' and 'wept'. The major problem that the two inventors were faced with, was how to develop the method of presenting the data to the network and in this case it was the verbs and their past tense.

A common example used in training a network is training the network to output 'on' when the input is 'off' and vice versa. Suppose there are 4 input units and 4 output units. The ANN will be trained on a number of samples and after several iterations it will know to output the desired value with the logical rule when the opposite value is presented. As an example of the learning ability, after the network is trained it is given the inputs 'off', 'off', 'on', 'on' in the 4 input PE's and the 4 output PE's would give the values 'on', 'on', 'off',

'off' respectively. The network succeeds in learning the logic without memorisation because it is presented with a pattern not seen before and is able to output the correct response.

The latter two examples demonstrate that the ANN—like the human brain—does not work like conventional computer software possessing a database of rules, rather using a dynamic learning process. This learning process makes us behave (output of ANN) in such ways that are difficult to define using logical rules.

The ANN research at this stage has focused on the electrical aspect of the behaviour of the nervous system neglecting the chemical aspect. In the brain chemicals involved in the operation of the neurotransmitters are used with success to treat brain disorders like schizophrenia. One of the assumptions is that they influence the brain by changing the connection strengths between the neurons. In an analogy to ANN, the chemicals are influencing the magnitude of the weights.

In ANN no meaning is attached to the dimensional topology in which the different PE's are arranged. It is meaningless to depict the ANN in two or three dimensions, it is the same ANN. On the contrary, in the brain the common belief is that there is an important role to the geometry of the neurons in the brain just as there is an importance in the position of lenses of an optical device, or the position of wheels in a clock. Neurons in the brain function just to either enhance or inhibit the firing of other neurons only, unlike neurons in ANN that enhance and inhibit simultaneously.

1.5 Comparison between ANN and regression analysis

ANN and regression have a number of similarities in that both are used to model relationships. However, they also differ in a number of ways. The two methods are compared below in the following areas: coefficients vs. weights, extrapolation, simplicity of the model, computation time, cross validation, model interpretation and optimisation respectively.

The coefficients in regression are analogues to the weights in ANN. The values of the coefficients are determined according to the least squares method. The values of the

weights in ANN are determined by the training process. (If biases are in use these are also adjusted through the learning process of training). It is not possible to calculate the regression equation when the number of coefficients exceeds the number of cases. In contrast, it is possible to train ANN when the number of weights and biases exceeds the number of cases.

It is not recommended to extrapolate beyond the values of the independent variables (Mendenhall & Sincich, 1996). This can lead to ridiculous results. In economics one can plot on the x-axis the advertisement expenditure and on the y-axis the sales. Extrapolation to a value of $x = 0$ could lead to a value of sales that is negative and this is physically impossible. In a tablet study, first presented in Chapter 3, using the method where data is partitioned into a training set and a small validation data set there is less chance of extrapolation. In the jacknife method, as described by Mendenhall & Sincich (1996), extrapolation is inevitable in some of the cases that are validated. The logic is: if one does not map a certain area, the behaviour of the function in that area is not known. With a number of independent variables, multi-dimensional plots are useful to determine whether one is extrapolating for any given values of the idependent variables. Looking at the values of each one of the independent variables separately it is not possible to know whether one is extrapolating or not. These multi-dimensional plots are also useful to examine areas where interpolation might be unreliable, i.e. where there are big 'holes' in the space that the independent variables create.

The model must be made as simple as possible. The model is built with as few coefficients/weights (and biases) as possible. In doing so the problem of overfitting is avoided (Masters, 1993). That means that the problem of a model learning any unimportant details in the data instead of learning the essential general trends is avoided. The disadvantage of this method is that the lower number of neurons may not be sufficient to characterise the data adequately. "Additional neurons increase the chance that even a local minimum will yield a low error." (Demuth & Beale, 1994). With fewer neurons there is less freedom in the network so the *error* surface that the function describes is less complicated and there is more chance for the network to be trapped in local minima with high error. In regression, by definition, if the number of coefficients is equal to the number of cases, the value of R^2 is equal to 1. In the latter case, we have a model in which 100% of the sample variation is explained by using x to predict y. But this model is probably bad for

prediction. The model could be memorising the data rather than generalising because it is too complicated. It is possible to construct a model with fewer coefficients with a lower R^2, with a good chance that it predicts better.

Coding means transforming a set of variables into a new set of variables (scaling the data). Coding is recommended for both methods: In regression it prevents the rounding error which occurs when multiplication of the matrix to calculate the coefficients takes place, and problems of multicollinearity (Mendenhall & Sincich, 1996). In ANN it helps the system to learn (Masters, 1993). For example, in this work, upon trying to train a network without coding, the ANN after a few iterations stops learning (the error does not decrease). After coding the same ANN architecture, the ANN was able to learn the data well.

In linear regression the same model will produce the same coefficients each time and the same prediction. In ANN it is possible to train the same network topology but result in a different prediction, because there is a random factor that may enter into the learning process (if the ANN is initialised with random weights) producing different weights and/or biases each time. With non-linear regression the situation is similar to the ANN.

The required computation time is a lot longer in ANN than in regression. Especially in the jacknife procedure, a computer goes through the jacknife procedure (see below) in regression in a few seconds whilst in ANN it takes the computer comparatively much longer to train.

Cross validation can be used to choose the best ANN topology/regression model. The data is partitioned into a training and a validation data set, the latter being used for testing the performance of the different topologies of networks that were trained or the different regression equations. The network with the smallest average percentage error is chosen as the best network (Murtoniemi et al., 1994a & 1994b). Murtoniemi et al., used cross validation for building the model (choosing the topology) as well as validating the chosen network in the same step. Another approach is to select the best topology of ANN by training different topologies on the entire data set and plotting a graph of the number of hidden neurons on the x-axis against root mean squared error (RMS) on the y-axis. Using the graph one can decide on the minimum number of neurons at which the RMS begins to stabilise. As the number of neurons increase the RMS as well as the slope of the graph

decreases. At this point one can use cross validation for validation of the selected ANN. One validation method is the leave-one-out method, or Jacknife, as in the work of Hussain et al. (1994). This is a special case of cross validation - the Jacknife can be used to detect an influential observation. Any observation for which the difference between the observed and the predicted value based on the model without that observation (this difference is called the deleted residual) is large relative to the predicted values is said to influence the regression fit (Mendenhall & Sincich, 1996). Once the observation is detected one can decide:

1. That there may have been a mistake in measuring that observation. This may involve tracking previous work.
2. To omit that observation.
3. To change the model.

Some of the greatest discoveries in science involved detecting observations that did not fit the current model, (e.g. quantum mechanics) and as a result another more appropriate model was established. It is especially relevant when one is interested in building the model not just for prediction but also for understanding the relationship between the parameters in the model. This leads to the next issue of how to deal with model interpretation.

To summarise, although terms like jacknife and leave-one-out method are used interchangeably in this work as well as other articles, jacknife is the name given to the leave-one-out method used for validation purposes and not in detecting influential observation(s). When the data is split into two sets, one for training and one for validation, (e.g. splitting the 27 data points into 22 for training and 5 for validation), this method is called split-sample validation (although in this thesis and in other articles this is referred to as cross-validation) as is the norm.

It is much more difficult to attach any physical meaning to the weights in ANN than the coefficients in regression analysis. Sometimes one can derive a simple regression equation that enables us to understand the system under investigation. In other cases it can be that it is not possible to interpret either the regression equation or the weights in ANN.

"Optimisation is the process of seeking the best solution for a system or activity." (Biles & Swain, 1980). The process entails manipulation of independent variables in order to get the

desirable response. Optimisation of the system is much more straightforward in ANN than in regression because one can see the entire system as a single entity. In the tablet study there is a network with 3 input neurons and 8 output neurons and the weights describe the relationship between them. One cannot, in our case, build a single regression equation that describes the relationship between the independent variables and all the response variables. It is possible to do network inversion in ANN: the user asks the net what properties an input pattern must have in order for the net to generate a specific output (Zell et al., 1994). The formulator in that case can ask the ANN what compaction force, lubricant level and disintegrant level the tablet must have in order to have appropriate dissolution rate and other response variables suitable for the pharmacopoeia constraints. If the researcher has an ANN with only one output, he/she can use the same optimisation procedures as in regression analysis. For example, plotting a contour plot for each response variable, marking the constraints based on same criteria, projecting the plots one above the other and finding the region of independent variables conditions that fulfils all the constraints (Schwartz, 1973).

1.6 Expert systems

Expert systems emulate the decision-making system of a human expert. Professor Edward Feigenbaum, an early pioneer of expert system technology, has defined an expert system as "an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution" (Feigenbaum, 1982). An expert is one who has expertise in a specific domain. An expert is one who solves problems that are unsolvable by one who has no knowledge in the expert's field or one who solves problems much more efficiently than another who is not an expert in the field.

It is common to use the term expert system in any system that uses the technology of expert systems. The technology can encompass computer language, programs and hardware. The knowledge in an expert system can come from human resources by experts in the fields or from literature.

The user of an expert system supplies input and is given advice by the system based on the information within the expert system. The expert system is made up of two parts: a

knowledge base and an inference engine. In response to a user query the system supplies answers.

Expert systems can also aid experts in the field by supplying solutions to difficult problems or reducing the time to get an answer. As these systems gather more and more information they improve and can give more help. With time the complete expert system evolves. Expert systems are suitable just for a specific domain just like any human expert who has expertise in a specific domain. The expert system unless programmed otherwise has no knowledge whatsoever on other domains.

In which cases are expert systems useful? If conventional programming can solve the problem then an expert system is not the best choice (Giarratano & Riley, 1994). Expert systems are generally designed very different from conventional programs because the problems they tackle usually have no algorithmic solution (Giarratano & Riley, 1994). Hence, if a simple program can solve a problem there is no need in using more complicated technology. Expert system must have a well-bounded domain for the problem it represents and what its capabilities should be. Sometimes solving problems in a certain domain requires knowledge from other domains, e.g. solving a medical problem might require delving into the chemistry and biochemistry and other domains as well. It is difficult to restrict the number of domains involved. When the problem-solving knowledge is mainly heuristic and uncertain it is appropriate to use an expert system.

1.7 Modelling, optimisation and expert systems in pharmaceutical development

The following paragraphs will explain in detail what has been done in the fields relating to this thesis and some notable studies have been reviewed critically.

1.7.1 Modelling and optimisation in previous studies

ANN have been used in all stages of pharmaceutical formulation development from trying to predict granule properties (Murtoniemi et al., 1994) followed by trying to predict tablet properties (Turkoglu et al., 1995), to optimising tablet coating (Turkoglu & Sakr, 1992) and finally the prediction of pharmacokinetic profiles by dissolution parameters (Hussain, 1997; Dowell et al., 1997 & 1999). Hussain was the first one to begin simulating pharmaceutical tablet properties with ANN (Hussain et al., 1991). He focused on the predictions of dissolution profiles of sustained release products (Hussain et al., 1991 & 1994) and he continued with *in vitro — in vivo* correlation of the dissolution and pharmacokinetic profiles (Dowell et al., 1997). These 3 stages in-process control, namely granule properties, the properties of the finished tablet and its pharmacokinetic profile are mandatory by many health authorities. The focus on the final stage is a natural continuation from previous studies and is the most expensive one. If tablet properties, such as dissolution profile are optimised, and the pharmacokinetic profile of the tablet does not meet the required criteria, it means loss of money to the pharmaceutical company (circa $30,000 based on personal knowledge of a bioavailability study done by the company I work for). These experiments (called bioequivalence for generic products) are expensive since they involve taking blood samples from healthy volunteers under hospital supervision. Certain health authorities do not require proof of bioequivalence for some generic products like paracetamol. In the following sections the modelling and optimisation of mainly solid dosage forms will be discussed for each of the stages of formulation development mentioned above.

1.7.1.1 Modelling (ANN & regression) and optimisation of granule properties

Both ANN and regression models have been used for the prediction of granule properties (Murtoniemi et al. 1994a). They used a fluidised bed granulator with a software package that controls the process. The independent variables that were controled were inlet air temperature, atomizing air pressure and binder solution amount. The measured responses were granule size and its friability. They performed 27 experiments in a full factorial design with 3 levels. A separate 5 experiments were performed to generate the validation data set. The models were built using the results of the 27 experiments and tested on the 5. Stepwise regression was used for building the regression equation. Although it is not mentioned which model they began this procedure with, it is suggested that they began with a second order equation. Average percentage error was used to evaluate the ability of the models to predict. The authors state that "the number of hidden layer neurons should be much lower than the number of training samples." However, the number of neurons in the hidden layer(s) could be much lower than the number of training samples and yet the number of weights could be more than the number of training samples. The number of weights relative to the number of training samples is the important factor that influences the generalisation ability. The ANN was fully connected with one or two hidden layers. The number of neurons ranged from 3 to 15. Hence, the number of weights in the ANN with 15 neurons in the hidden layer is $3 \times 15 \times 2 = 90$ (without taking into account biases that are a special class of weights) and this is obviously more than the number of data points. One opinion regarding this subject (Hagan et al., 1996) is that "For a network to be able to generalize, it should have fewer parameters than there are data points in the training set". It is interesting to note that the best ANN chosen for both responses had more weights than the number of data points, since the chosen topologies were with one hidden layer that consisted of 11 neurons for modelling granule size response, and 12 neurons for modelling granule friability response. The authors talk about generalisation ability and how to improve it when using ANN models but it is not clear how they arrived at the conclusion that the trained ANN had generalisation ability! The average value of granule size (of the 27 experiments used to build the model) is 370 μm. Using this value to predict the validation set yields 12.41 average percentage error, which is less than the 14.71 average percentage error arrived at by the authors (for the best ANN model). As will be seen later sometimes ANN or regression can not generalise well. In the study described in Chapter 5 the

ANN/regression models generalised well for 5 of the 9 response variables. Another problem is that a set of 5 cases is most likely too small to determine generalisation ability (the probability that it can represent the full population is low). In the study of Murtoniemi et al. the training end point of the ANN was also varied. In a second study by Murtoniemi et al. (1994b) modelling of granule flow rate was included in the regression and ANN modelling. The ANN was modelled with just one hidden layer but with up to 25 neurons in the hidden layer and with more training end points. The authors also repeated the modelling of ANN with this method for the responses of mean granule size and granule friability with the aim of improving the results.

Takayama & Nagai (1989) tried to optimise indomethacin release from granules. The independent variables manipulated were concentration of polyvinylpyrrolidone (PVP), concentration of carboxyvinyl polymer (CP) and addition rate of PVP solution to indomethacin suspended CP solution. The responses modelled were time to release 50% of indomethacin from powder, time to release 50% of indomethacin from granules, moisture uptake of powders, indomethacin content of powder and sample recovery. The 3 independent variables were at 5 levels, 2^3 cases dedicated to full 2 level factorial and the rest for the other 3 levels. In the end there were 15 experiments at 5 levels. They used a second order polynomial with 10 parameters just like Schwartz et al. (1973). Unlike Schwartz et al., Takayama & Nagai (1989) selected the best factors by choosing the equation that gave the best coefficient of correlation. This correlation coefficient took into account the degrees of freedom. In total there were 511 cases (2^9-1). To summarise, this approach selects the best regression equation from all possible combinations using the R^2 value and taking into account the number of coefficients (which determines the degrees of freedom). For example, if for 3 coefficients the value is 0.85 and for 6 coefficients the value is 0.87, the equation chosen will be the one with 3 coefficients. According to Bolton et al., 1997, stepwise regression is considered a better method than "all possible regressions" approach described here. The optimisation in Takayama & Nagai, 1989, study was done using a Monte Carlo approach that uses random number techniques to search for the optimum solution.

1.7.1.2 The use of ANN and regression to model powder properties

Bourquin et al., published several articles on modelling using ANN focusing on different responses to those discussed up till now. Bourquin et al. (1998c) tried to model an in-process system. As opposed to the granule modelling described earlier (Murtoniemi et al. 1994a & 1994b) here the in-process mixture was in its powder form. The properties of the powder are important for the success of direct compression process. The responses modelled were powder flow, tap and bulk density. Also modelled were α (alpha) and β (beta) parameters which relate to the following equation:

$N / VR = \alpha * N + \beta$ (Yamashiro et al., 1983)

Where N is the number of taps and VR is the volume reduction. The input variables that were manipulated were concentrations of silica aerogel, microcrystalline cellulose, magnesium stearate, low substituted sodium carboxymethylcellulose. The network was trained using backpropagation with hyperbolic tangent function as the activation function in both layers. The data was split into 14 data points for training and 3 for validation. The training was stopped as soon as the MSE of the validation set began to increase. The Mean Square Error, MSE is equal to SSE/n (where n is the number of data points). The ANN was trained 10 times each time with new random weights. The network/regression model showing the best R^2 of each response was then selected for the model comparison. The regression models were trained on all the data using the maximum R^2 to avoid overfitting. The conclusions of this research were that ANN and regression gave comparable fitting results but regressison is superior in modelling outliers.

Bourquin et al. (1998b) explored the influence of 6 input variables that were concentrations of silica aerogel, microcrystalline cellulose, magnesium stearate and carboxymethylcellulose as well as compression force and dwell time, on the responses of normalised ejection and residual forces. One hundred and two experiments were done overall. They were split into 87 for training and 15 for test set. The MSE in predicting the test data set was monitored and the training was stopped as soon as this value no longer decreased but began to increase (the start of overfitting). The network was trained 10 times, each time using new random set of initial network's weights. The network showing the best fit (smallest MSE) was then selected for the model comparison. The split for the comparison was done differently from the selection of the best ANN: 80 samples for

learning and a validation data set of 22 samples randomly chosen. The models were developed as described above but utilising the reduced training data set (i.e. statistical response surface modelling with 80 samples and ANN modelling with 68 samples as training and 12 as test data set). Since Bourquin et al. .(1998b) had enough data they could split the data into training and validation set with a reasonable probability that their validation set of 22 samples represented the population it came from. They used a simple backpropagation algorithm for training but in addition to the regular connections of multilayer perceptron (MLP) network, there were also connections from the input layer directly to the output layer (they were fully connected). Bourquin et al. (1998b) termed this type of ANN "generalised feed forward multi-layer perceptron network" (GFF-MLP). They began to test the optimum topology for the number of hidden neurons beginning with the maximum number of hidden neurons as defined by Kolmogorov's theorem and then reduced the number of hidden neurons until a generalising working ANN was obtained. This strategy had been suggested earlier by Maren et al. (1991). Kolmogorov's theorem states the minimum number of hidden neurons should be twice the number of independent variables plus one to compute any arbitrary continuous function (Hecht-Nielsen, 1987). They stated in this study that six parameters give 13 hidden neurons that are enough to model their specific problem according to Kolmogorov's theorem, and that the neurons were reduced to 12 units in order to reduce the risk of overfitting. It is the opposite of the approach of beginning with the minimum number of hidden neurons and adding one neuron each time till generating ANN which can learn and generalise (Masters, 1993). The approach of Bourquin et al. is an acceptable method described in the literature (Maren et al., 1991).

1.7.1.3 Modelling (ANN & regression) and optimisation of tablet properties

Bourquin et al. (1997b) modelled the influence of several variables on tablet crushing strength (N), percentage of drug dissolved after 15 minutes (%) and time to 50% drug dissolution (min). The variables were: formulation factor (the drug was granulated using two different formulations), 3 levels of matrix filling speed (rpm), 3 levels of precompression force (kN), 3 levels of compression force (kN) and 3 levels of rotation speed (1/hr). Two different ANN types were examined, GFF-MLP and self-organising feature map MLP (SOFM-MLP) for modelling this data. GFF-MLP has 'short cut' connection between the input layer and the output layer (this type of ANN was used in the article described earlier). Also in the output layer of the GFF-MLP and SOFM-MLP there is a hyperbolic tangent activation function. These ANN were trained with backpropagation. The regression model chosen was a polynomial of third order including all interactions. It was not mentioned if other regression models were investigated or if a method of variable selection was employed, which is a reasonable thing to do in order to reduce this large equation with many parameters. The training of the models was on a reduced data set, and validation used all the data. This approach is unique in the literature. The approach seems to impose the problem that there will be better validation results for the data points that the models had been trained on, since the models could memorise these data points. No ANN or regression model was able to predict the two dissolution responses since in this author's opinion, some additional parameters influence these responses.

Bourquin et al. (1998a) tried to model tablet properties of tensile strength, friability, disintegration time and dissolution profile (percentage of drug dissolved after 15, 30, 45 and 60 minutes). The input variables that were manipulated were: percent of silica aerogel, microcrystalline cellulose, magnesium stearate and sodium carboxymethylcellulose. As well as dwell time and compression force. ANN models were compared to regression models. The type of ANN was GFF-MLP with the hyperbolic tangent activation function in both layers and the training method was backpropagation. The ANN was chosen from ten ANN with different initial random weights. 28 data points out of 205 were chosen. Using the 28 data points the ANN was fed with 24 data points for training and 4 data points were left to test. When the MSE of the predictions regarding the test set started to increase the training was stopped. The network showing the best overall fit (R^2 for each response) was

then selected for comparison of the models. The authors stated that as long as the achieved R^2 coefficients does not surpass the value of maximum R^2 there is no overfitting of the model. This value of maximum R^2 was calculated from the averaged variances of the repeated measurements and from the variance of the means. The use of 14% out of 205 data points for training and the rest for validation was employed to compare between the ANN and regression models. Two comparison experiments were performed, in one the training set was chosen using an organised design and in the second the training set was selected in a random way. This research concluded that ANN methodology is much less sensitive to organisational level of trial design than the response surface methodology (RSM) and is therefore better adapted for the data analysis of the results of historical or poorly organised trials. This issue of which method predicts better when the experiments are not from a well-organised design is addressed in the current thesis.

Turkoglu et al. (1995) used 27 experiments in a full factorial design. Twenty-two experiments were used for training and five were separated from the data set for validation. This design of experiments and validation method was used in one of the studies in Chapter 3 of this thesis. The input parameters used were lubricant type, compression pressure and mixing time. The 3 responses modelled were crushing strength and percent released after 30 and 60 minutes. They compared ANN to a second order regression equation. There was no minimisation of these equations. Nevertheless, separate equations were built for each lubricant type, so although they were second order they included only 5 variable terms. The ANN were optimised using 'network reversal'. They state that the network reversal function predicts appropriate input settings needed to achieve desired output settings. There was no elaboration of this method. This method of optimisation is also mentioned by Achanta et al. (1995) where it was stated that this is an advantage of ANN since they can perform 'inversion' for purposes of optimization. A method of directly optimising ANN (although not termed as an optimisation method) was mentioned in a review of ANN applications in chemistry (Sumpter et al., 1994). The goal of the method is to find ideal input examples to specified outputs. In this method no solution can be found but a set of good solutions can be found. This research uses multiobjective optimisation not directly from ANN but using multiobjective optimisation routine with ANN. The same concept of optimisation, as employed in this study, is described in a review about optimisation using ANN (Takayama et al., 1999). This review describes optimisation of ANN trained to model a transdermal therapeutic system. The aim of the latter study was that the drugs from

this system will penetrate efficiently into the systematic circulation to achieve sufficient concentration for the desired therapeutic effect. The input parameters were the amount of ethanol as one variable and the amount of ortho-ethylmenthol as the second one. The responses measured were apparent penetration rate, lag time and total irritation of the skin score. This article also mentioned another 2 studies that used similar optimisation techniques (Takahara et al., 1997a & 1997b). A completely different approach to optimisation is presented by Shek et al. (1980). In this method there is no training of ANN/regression equation prior to optimisation but it is done dynamically. The researcher feeds the optimisation routine with a number, and after a calculation based on the principle of trying to get far as possible from the worst point one gets a suggestion from the computer regarding the next independent parameters one should use. The system to optimise was a capsule formulation problem. There were 4 independent variables: percent of drug substance, total capsule weight, disintegrant and lubricant levels. And 3 responses: rate of packing down, percent dissolution after 8 and 30 minutes. The score for each set of the 3 responses was done by the linear combination of the 3 weighted responses. The weights for the responses were given by their importance. The packing down rate was given weight of 50%, dissolution at 30 minutes was given importance of 40% and dissolution at 8 minutes was given importance of 10%. So the equation that measured the success of each experiment looked like:

$$R_t = 0.5 \times R_1 + 0.4 \times R_2 + 0.1 \times R_3$$

Where R_t is the objective function and the different $R_{i(1\text{-}3)}$ relate to the responses discussed previously. This concept of giving weights in optimisation of several responses simultaneously is common in many multiobjective optimisation routines. The ability to give appropriate weights depends on the expertise in pharmaceutical formulation of the one who performs the optimisation.

Bohidar et al. (1979) modelled tablet response variables using also the stepwise regression technique which was used by Murtoniemi et al. and also was used in this thesis. They examined 5 input variables to model 10 responses that are similar to the tablet study in this thesis. They found compression pressure and lubricant level to be the most important. This work uses these two factors as well as the disintegrant level. Bohidar et al. used 27 tablet formulations. Schwartz et al. (1973) also modelled tablet properties using 5 input variables to model 8 responses similar to the ones measured in this thesis. Schwartz et al. did not use a method of variable selection but used an equation of 21 terms to model 27 experiments in

a second order model. Hence, allowing more chance for the regression equation to generalise and thus better than Hussain et al. (1991) that used 15 variables to model regression equation with 15 coefficients. Schwartz et al. also used numerical as well as graphical techniques in multiobjective optimisation. This work uses multiobjective optimisation but with ANN instead of regression equation and also with a more modern method of optimisation. Hussain and co-workers continued research on predicting the dissolution profile using ANN this time from a tablet matrix with leave-one-out validation (Hussain et al., 1994). The independent variables were molecular weight, intrinsic dissolution rate, pKa, salt type, drug to polymer ratio and hydration rate of the polymer. The responses measured were percent drug released at 1, 3, 6 and 12 hours. They optimised the number of hidden neurons but also the number of epochs. The different ANN were trained on the training set and the one with the minimum root mean square error was chosen (also with graphical method that will be explained later). The leave-one-out method was done on the selected ANN. Regression presents no problem in term of training time for the leave-one-out method but in ANN it is time-consuming. On the other hand when there are few data points it is the most reliable way of validating the model. Hence finding ways to estimate ANN leave-one-out results are under research today (Myles et al., 1997) but there is as yet no method that one can confidently say predicts the results of this validation method.

1.7.1.4 Modelling (ANN & regression) and optimisation of *in-vitro* drug release profile

Another important part of process modelling and optimisation is tablet coating. Turkoglu et al. (1992) checked the influence of the following fluidized-bed coating parameters: polymer amount, coating temperature and spray nozzle pressure. The responses characterised drug (theophylline) release profile. The regression equation was second order model with ten parameters. Optimisation was done using graphical methods and with simplex algorithm. There was no method of variable selection and there was no validation of the model. Turkoglu et al. (1992) did simple maximisation of the theophylline release. The goal of their maximisation problem was to get more than 80% theophylline release after 11 hours. Since there are 3 regression models for the release profile after 5, 8 and 11 hours, optimisation of these three responses simultaneously according to desired release profile seems to be a more appropriate approach that is in accordance with the approach to multiobjective optimisation in the current thesis. After optimisation, a batch with the

optimised tablet coating parameters was manufactured. The batch gave more than 91% theophylline release after 12 hours. This can be given as an indication that the model generated was adequate—since the optimisation routine done on the model gave an optimum result which was validated with real life experimental result.

Hussain et al. (1991) characterised the release exponent and the dissolution half time from a capsule matrix. The input parameters varied were four formulation variables: carboxymethyl cellulose, hydroxypropylmethyl cellulose, hydroxypropyl cellulose and hydroxyethyl cellulose. They trained the ANN/regression models on 15 formulations and used a validation set of 8 formulations. The topology of the ANN was 4 input neurons, 2 output ones and the number of hidden neurons was varied from 4 to 14 nodes. The best ANN selected was the one that gave the lowest sum squared error for the training set. The selected ANN was with 8 hidden neurons which give $4 \times 8 \times 2 = 64$ weights and this is more than the number of data points. Looking at the predicted values by ANN and regression show that ANN predicted better all dissolution half life results and regarding the release exponent regression gave just two predictions that were better than the ANN predictions, so it is obvious ANN modelled the data better. It is a puzzling fact (as well as in others) that they used several ANN models but just one regression model. Furthermore, the regression model used had 15 coefficients (on 15 data points!) reducing the probability of generalisation to the regression equation. They used a first order regression equation but with all interaction terms added. The high R^2 values of 0.989 and 0.976 for the regression models for the two responses suggest there is a problem of overfitting to the training set. The selection by the authors of a long equation without a method of variable selection is not appropriate because it does not give a chance to generalise since there are too many variables.

Johnson et al. (1990) researched the release of drug from a controlled release matrix. The independent variables were component fractions of formulation constituents and the responses characterised the drug release profile. The statistical design of the experiments was composed of 8 extreme vertices (there were 4 independendent variables). These 8 extremes defined the geometric boundaries of a seven-sided hyperpolyhedron. Eight additional points were put at the centre of each of the 7 faces and one at the geometric centre of the mixture space, designated as the centroid point. The validation of the model was done by choosing 5 points randomly from the space within the extreme vertices. The

average difference between the drug release profiles predicted by the model and the profiles determined experimentally was 2.05%, which is well within the experimental error (4.5%). The regression equation was optimised by stepwise regression using SPSS. The F-test ratio for including a regressor into the equation was initially set at 0.01 (F_{in} = 0.01) and the F-test ratio for deleting or excluding a regressor from the model was set at 0.005 (F_{out}=0.005). This thesis also used SPSS software for stepwise regression and the tuning of the F-test ratio for including or excluding regressors will be discussed later.

Ebube et al., 1997, tried to model hypothetically eleven controlled-release formulations. The input variables manipulated were fractions of two polymers and the response measured was percent of drug dissolved after one hour. The leave-one-out method was the validation method. Examination of the predicting ability using this validation method shows that the error is big for the slowest dissolved tablet and there is no ability to extrapolate. However, for the fastest dissolving tablet it was possible to extrapolate efficiently.

1.7.1.5 Modelling with ANN and optimisation of a pharmacokinetic drug profile

Finally, the last stage in formulation development is attaining the desired pharmacokinetic profile. Previously, prediction of dissolution profile was demonstrated, the link between this and the prediction of pharmacokinetic profile will now be discussed. The use of ANN for predicting pharmcokinetic/pharmacodynamic parameters is a well developed area (Hussain et al., 1992; Veng-Pedersen & Modi, 1993; Brier et al., 1995; Jogarao et al, 1995; Jogarao et al, 1996). There is a good introduction to this field by Erb (1995). This section will describe just one part of this subject that relates to pharmaceutical solid dosage forms.

Not many studies are in this new area of *in vitro-in vivo* correlation using ANN (Dowell et al., 1997/1999; Hussain, 1997). The ones by Dowell et al. will be discussed here (the two articles are on the same study). This study demonstrated the issue that the same data can be presented in a different manner to different ANN. To obtain the dissolution profile the percent dissolved at 7 different time points was measured (input variables). To obtain the pharmacokinetic profile (response variables) the plasma concentration at 15 different time points was sampled. 9 subjects were sampled for their pharmacokinetic data and 6 samples for each point in the dissolution profile. There were 4 ways of associating the data (input-

output pair) and each is suited to a different ANN. Association 1 receives as input the 7 dissolution points and as output 15 pharmacokinetic points, these represent the dissolution and pharmacokinetic profiles respectively. Overall they had 6×9=54 data points for each formulation. In Association 2 they used ANN with topology of 8 input units and 1 output unit. The first input represented the pharmacokinetic time and the output is the corresponding pharmacokinetic output. The other 7 input units represent the entire dissolution profile. Overall they had 6×9×15=810 data points for each formulation. Association 3 was represented with ANN of 2 input units and one output unit. The first input unit represented the dissolution time, the second one represented the corresponding data point in the dissolution profile. The output unit represented the corresponding pharmacokinetic data point suitable for that time point. Overall they had 6×9×7=378 data points for each formulation. Association 4 used the same topology of Association 2 of 8 input units and one output unit. One input unit represented the pharmacokinetic time. The other 7 input units represented the dissolution data points till that time point, and the other input units got the value of zero. In this manner as the pharmacokinetic time of sampling is progressing there are more values in the input units that are not zero represented by real values from the experimental data of dissolution. The output unit is the pharmcokinetic data corresponding to the time of the input unit. Overall they had 6×9×15=810 data points for each formulation just as in Association 2. This study concluded that ANN are capable of predicting pharmacokinetic profile based on dissolution profile. The next step the study dealt with was to run an optimisation routine to get the optimised dissolution profile that yields the optimum pharmacokinetic profile. For that purpose they used a technique called genetic algorithm, which works by evaluating the pharmacokinetic profiles according to many dissolution profiles and retrieving the best solution.

1.7.2 Expert systems in pharmaceutical product formulation development

Expert systems for product formulation are in use in many fields like agrochemicals, alloys, printing inks, resins and varnishes, lubricating oils, vinyl blowing agents (Rowe & Upjohn, 1993). Pharmaceutical expert systems are in use in various fields: parenteral development (Rowe et al. 1995), skin care products (Wood, 1991), and in the field of solid dosage form development that will be discussed extensively in the following sections.

Lai et al. (1996) developed an expert system utilising more than 2000 references relating to problems in capsule filling and related problems in powder technology. This database is permanently updated. A second database provides information on excipients. The database contains 750 different formulations of 250 drug substances. A group of 10 experts in the field of capsule formulation was employed to generate many facts and rules. The formulations are deduced automatically. The programming language used was C.

Rowe et al. (1997) surveyed expert system shells that can be used to develop pharmaceutical formulation expert systems. They mention several technologies that are incorporated into expert systems like ANN or genetic algorithm. ANN and genetic algorithm are incorporated into Cad/Chem which is a software used to build models and optimise formulations (Colbourn & Rowe, 1996). The use of the software is demonstrated by Kesavan & Peck (1995). Rowe & Upjohn (1993a) used an expert system to identify and solve film-coating defects. They developed this with an expert system shell.

There are several tablet formulation expert systems (Podczeck, 1992; Ramani et al., 1992; Rowe & Upjohn, 1993b; Stricker et al. 1991 & 1994). Rowe and Upjohn (1993b) described a Zeneca expert system for tablet formulation. The flow of the program will be described in brief. Drug properties are entered into the database. Then the user selects a tablet profile, accordingly the system automatically selects the formulation excipients. It selects the excipients that are characterised by their functional category groups. The functional category groups in the Zeneca expert system are: filler, disintegrant, binder, surfactant, glidant and lubricant. It has a database on excipients. Afterwards, the Zeneca expert system predicts the tablet properties that the suggested formulation will have. Then the user enters the measured tablet property results like disintegration time etc. The Zeneca

expert system then compares the predicted results to the observed ones and accordingly it gives advice that the user can accept/reject. According to the user input in response to the advice, the system suggests formulation changes. This iterative process, of suggesting formulation, feed the observed tablet properties, give advice and suggesting a new formulation, is repeated till the desired formulation is achieved.

The Cadila expert system (Ramani et al., 1992) is written in Prolog. Prolog is a programming language specialized in the field of artificial intelligence and as such programming expert systems is one of its domains (Bratko, 1990). The system selects the appropriate excipients according to facts like interactions between drug and excipients. There is a rule embedded in this system that all formulations must contain one binder, one disintegrant and one lubricant. Excipients related to other functional groups are added if necessary. The user can accept or reject the purposed formulation. In the latter case the system will suggest another formulation. The Cadila expert system has no process of formulation optimisation.

Podczeck (1992) developed a knowledge based system for the development of tablets. Physical, physicochemical and pharmaceutical measuring methods were used to characterise 15 drugs in terms of their pharmaceutical behaviour. Each of these drugs was mixed with several excipients and the properties of these drugs in the mixtures were measured. The relationship between drug properties and the behaviour of mixtures of drug substances and excipients was done using canonical analysis, which is a multivariate statistical method. Hogan et al. (1996) also used this type of statistical analysis and the capsule data from their article was used in this thesis. The relationships found were calculated as facts and the rules of knowledge based system. The expert system was validated successfully using another set of 5 drug substances.

The Galenical Development System, Heidelberg (GSH) has been designed for the development of direct compression tablets, hard shell capsules, aerosols and intravenous injection solutions. It uses functional groups for the excipients and has a database with important information like compatibility of the excipients. It uses functional groups also for the chemical and physical properties of the active ingredients, manufacturing processes and other types of data. The desired properties of the formulation are set for the GSH and the system suggests a formulation according to a rule-based mechanism. If the suggested

formulation does not satisifies the desired properties the development continues and the system suggests another formulation. This system has a back-tracking mechanism to go back to the previous step or abort. The programming language of this expert system is SMALLTALK V. More details about this expert system can be found in the book 'Intelligent Software for Product Formulation' (Rowe & Roberts, 1998). There are two articles about this system (Stricker et al. 1991 & 1994) but these are in the German language with no translation available through the British library.

To summarise the evolution in methodology, in regression field there was an evolution from equations that are usually of second order type without variable selection to final equations that passed variable selection processes like stepwise regression. This evolution is not seen in a chronological order. For example, Hussain et al. (1991) did not do variable selection although other researchers like Bohidar et al. (1979) did it before. In the field of ANN it can be seen that as more and more studies are performed more parameters are varied in ANN, like changing criteria to stop training as the error goal or the number of iterations. Parameters that concern ANN topology like the number of hidden neurons and the connections between the neurons inside ANN were also varied. Other parameters that were mentioned are the type of activation function in the neurons of the hidden or output layer. Optimisation methods have evolved from graphical methods (Schwartz et al., 1973) to numerical optimisation methods (Takahara et al., 1997a & 1997b). Expert systems were discussed in the final stage since they are new to pharmacy and since they could incorporate regression, ANN and optimisation methodologies.

1.7.3 Present study

This part will be a brief summary of objectives and how this study differs from previous ones. Also, there will be a brief explanation of the Expha expert system contribution. It will emphasise points already mentioned in the review of previous studies.

This study attempts to investigate rigorously applications of regression analysis and ANN to the formulation and manufacture of solid dosage forms. Specificially it will use data from well-designed experiments (full factorial) of manipulating tablet formulation and process variables. On the other hand, it also will use limited data of manipulated capsule

formulation variables. The prediction of ANN and regression models will be compared using extensive statistical analysis that will focus on various aspects. There is an attempt to develop an expert system that helps novice formulators in formulation and for educational purposes for undergraduate and postgraduate students in pharmaceutical sciences. It will have rules, a database and also modelling with optimisation facilities.

The work which is presented here is different from previous ones (i.e. Hussain et al., 1994; Murtoniemi et al., 1994a & 1994b) in several points. Several regression models were used here whereas most use just one regression model. This study used different methods of training ANN while others used just one method of training ANN. This study used other learning techniques apart from backpropagation while others (excluding Bourquin et al. 1997b who also used a Kohonen network) used just backpropagation for training . After the derivation of ANN/regression models, and generation of the prediction results the study does not just present the results in terms like relative error in percent, but also used statistical techniques to see if the difference in the predicting ability is statistically different, a task that other studies have not done.

Today the advanced pharmaceutical formulator who is up to date with the software can use separate systems to aid him in formulation. An approach is using an expert system like PFES (Rowe, 1997) which suggests a formulation; after creating the formulation the results of the response variables are fed back to obtain a better formulation. The process of trial and error continues using the expert system suggestions till a satisfactory solution is found. A second approach is to use the expert system just for the initial formulation and then on the basis of this formulation conduct designed experiments and use ANN/Regression for modelling. These experiments are combined with optimisation tools that are within commercial packages like CAD/Chem (Colbourn & Rowe, 1996). One can also invent the initial formulation and design experiments and modelling of data with statistical tools like regression and ANN available on commercial programs. Part of the data modelling software available on the market has also features that assist in the final stage, which is optimisation of a formulation. In the Expha expert system the ANN/regression models are incorporated with a pharmaceutical database for decision making in the formulation process. This approach of using various modelling and optimisation techniques, which the user can control, in combination with heuristic rules applied to the database, creates a more powerful decision tool than each one of the methods alone, and this was not done before.

This work helps the formulator abandon the passive part of using an expert system, instead it helps in knowing what the expert system consists of. There is no process of building an expert system and implementing it from the expert to the software engineer and then to the programmer (which is probably best from the management point of view of getting the best expert system in the minimum amount of time). The systems built in the latter manner are protected and are not transparent and so the formulator has difficulty in using them from an academic point of view. They were built more as boxes that give solutions sometimes with the reasons for choosing the solution but as to how the system generated this solution is unclear.

Expha can help in learning and training of novice pharmaceutical researchers in the pharmaceutical formulation field. It can give the students the fundamentals of formulation development. It can educate them to try different models. And give them tools also to judge which is better. This process will force them to delve into statistics, and their use of this will enhance their judgment ability as researchers. It can also be of use to pharmaceutical researchers in industry as a tool for developing new and improving old formulations.

If the people using these systems want to delve into the world of database systems this work can elucidate this subject by examining Expha which is a transparent expert system. Alternatively they can decide that they want to analyse their knowledge and to implement their own heuristic rules. But before that they will have to create their own relevant tables of data. All this analysis of their decision making, and the data they use, will cause them to understand better their formulation problems and hopefully solve them better.

1.8 The structure of the thesis

Chapter 2 is the Background chapter, the following three chapters relate to the subject of building ANN/regression models and doing comparison between the best ANN and regression models. Chapter 3 is based on experiments with a full factorial statistical design. The following chapter deals with improving ANN modelling to this data with more emphasis on robust comparison between ANN and regression. A second set of data utilised had significant gaps in the mapping of the response space. Hence, Chapter 5 is about

modelling of limited data. Chapter 6 presents the subject of multiobjective optimisation and uses the experimental data used in Chapters 3 & 4. In order to demonstrate how ANN/ regression models are used through the process of data collection and decision making in the field of tablet formulation, they were incorporated into an expert system. This is dealt with in Chapter 7 as well as the incorporation of database and heuristic rules in the Expha expert system. Finally, Chapter 8 presents a brief summary and the main conclusions derived from the studies of this thesis.

2. Background

2.1 Introduction

This chapter begins with a description of tablet technology concepts such as formulation considerations and stability studies. Some of these subjects like the flow of powders also apply to capsules. After the domain of the problem is described, the modelling techniques to solve tablet problems follow, beginning with regression and then by background on ANN. The section on ANN will present the basic principles of ANN to give a better understanding of them. The backpropagation algorithm that is the most common ANN training method will be presented with variations to this algorithm. Radial basis function (RBF) ANN will be presented since in certain cases these are superior to ANN trained by backpropagation as is shown later in this thesis. Finally, expert systems, technology that sometimes incorporates the modelling techniques, will be discussed briefly.

2.2 Tablet technology

Tablets are the most popular dosage form. Tablets gained their popularity for several reasons. They are light and compact so are very easy to carry in a bag, for example (thus enhancing patient compliance), and have low transport cost from the factory to the patient. Another reason there is good compliance with tablets is that each tablet is a unit dosage form. In that respect, they are usually designed so that the patient takes one or two tablets as and when necessary, e.g. take one or two paracetamol tablets in a case of headache or one tablet of Aspirin® every day. Tablets are a solid dosage form and as such are considered quite stable relative to the liquid dosage forms. Tablets are also less prone to microbial contamination relative to liquid dosage forms. To meet pharmacological or market needs their performance can be manipulated, for example change of dosage from

two conventional release tablets a day to one sustained release tablet a day, in order to enhance patient compliance whilst maintaining the desired pharmacokinetic profile necessary to achieve adequate pharmacological action. A major advantage of tablets is their low production costs. Nevertheless, tablet manufacturing can be a very challenging task.

For the success of tablet manufacturing there are demands from the physical characteristics of the powder and also from the tablet press design. The tablet manufacturing machines today can work very fast, and this high speed could cause uniformity of weight problems. For uniformity of weight to be consistent the bulk powder poured into the die should always be of the same volume and homogeneity. The tablet should comply with several characteristics (responses) that are sometimes contradictory in their nature, e.g. the tablet should be strong enough to endure its travelling from the manufacturing site to the client and also to dissolve fast enough in the body.

The manufacturing process affects the tablet formulation whether this is granulation with water/solvent or direct compression. The process selection is dependent upon the drug characteristics such as sensitivity to heat and moisture, the flow characteristics of the mixture (drug/s plus excipients) and if there is a tendency for segregation. Granulation is required to remove dependency on physico-chemical characteristics of drug, improve flow, improve compressibility (enable the tablet to compact easier), reduce potential for segregation, increase density and improve worker safety by reducing dust hazards. There are two main granulation methods, dry and wet granulation.

In a dry granulation method two processes are commonly used. In the first called *slugging* the powder mixture is blended, then a heavy-duty tablet machine makes very large tablets (slugs). The tablets are broken down and screened and the mixture is placed in a regular tablet machine. In the second method, roller compaction, the mixture is placed inside a hopper, then a rolling auger force feeds the powder through two pressure rolls and the compacted material is due to the roller compaction. This compacted material, as in the first method, is milled and goes into a regular tabletting machine. The process variables that can be monitored in the latter process are roll pressure, roll speed, auger speed (powder feed) and roll profile.

In a wet granulation method there are two types of manufacturing, there is a traditional process and a modern one. In the traditional process there is a mixing stage in a planetary mixer for 30 to 45 minutes, followed by granulation for 15-45 minutes. The wet mass is then screened, a process that takes about half an hour to one hour. This is dried on special trays in an oven for about 16 hours then the granules are dry screened, a process that takes about half an hour. In the modern process using a high speed granulator with a fluid bed drier the mixing stage takes from one to five minutes, granulation from 2 to 5 minutes and the drying stage takes from half an hour to one hour. It is also possible to complete the process in two to three hours using a fluid bed granulator. Wet granulation can improve mixing of potent drugs or colours by incorporating the drug or the colour in the binder solution. Since the wetting is much better in wet granulation, dissolution rate in wet granulation can be much better than in direct compression. The disadvantages of wet granulation processes are that they are lengthy, costly and require the drug to be heat and solvent stable. After the granules are made, it is beneficial to test the granules in order to trace problems before the tabletting stage. In-process control (IPC) tests are also becoming a requirement by the regulatory authorities. The granule parameters that can be tested are shape, size, size distribution, density, surface area (related to porosity), consolidation and compaction properties of the granules, strength, friability and flow properties.

In a direct compression process, after the powders are mixed there is the tablet compression stage. Some pharmaceutical companies encourage direct compression processes in their tablet development guidelines because of the lower costs and easier process validation compared to the granulation process. The selection of process is directly related to the selection of excipients and drug form, e.g. paracetamol DC could be in the form of paracetamol coated with PVP (combination of paracetamol 96% and 4% PVP can be bought from ATABAY) to enhance its compactability and flow properties, so it can be used in a direct a compression process.

After the tablet is made there are several problems that could arise. Poor weight uniformity can be caused by poor flow of the powder used in the tablet compression machine. Inappropriate mixing can cause poor content uniformity. Mixing has its own methodology and it is not a simple subject, mixing for too long a time is also not good. The tablet may be weak (hardness test) which may be caused by an inappropriate use of binder. The tablets could be friable, meaning there might be problems in the transport from factory to the

patient and that coating of the tablets will be beset with major problems (if there is a friabiltiy problem before the tablet coating stage). 'Picking' (explained later) could be caused by lack of antiadherents. 'Capping' and 'lamination' could be caused by: elastic recovery, residual radial forces, presence of moisture, air entrapment, or machine problems like non-homogeneity of the force transmitted to the powder causing different densities in different areas of the tablet, a wear ring in the tablet machine die or damaged punches. Where there is a problem with the dissolution rate, it might be beneficial to examine the disintegration rate of the tablets. For tablets that do not disintegrate easily this may be due to the tablet hardness being too high or the tablets becoming a sticky mass in the presence of water. The appropriate choice of disintegrant and/or binder should solve these problems. For cases where there is no disintegration problem, the low rate of dissolution may be caused by too much lubricant (since most lubricants are hydrophobic) impairing drug dissolution. A sparingly soluble drug might exhibit low dissolution rate. Use of wetting agents can aid in solving this problem. As is demonstrated in this section, the key for solving many tablet problems is the appropriate choice of excipients. Hence, the use of excipients will be discussed in the following sections.

The drug is mixed with several other ingredients to make a tablet. These ingredients termed excipients help the tablet perform as specified. The excipients are divided into several groups according to their role. One excipient can belong to several groups since it has several roles and for each one of the excipient roles there is a different recommended concentration. The tablet does not have to contain all the excipient groups. On the contrary, the simpler the formulation the better, as more ingredients can cause more problems. No one excipient group must be included. Even the most common groups like disintegrants are not always required as in cases where both drug and tablet are easily dissolved in body fluids. A description of several important excipient groups: diluents, binders, disintegrants, lubricants, glidants and antiadherents follows. Excipient groups with specialised functions like antioxidants for ensuring drug stability or wetting agents for improving drug solubility (influence dissolution rate and bioavailability) will not be discussed. Other groups that will not be mentioned are absorbents, colours and flavours.

The most important ingredient in the tablet is the drug. The larger the amount of drug inside the tablet is the more dominant the drug properties are and should be considered. One of the important drug characteristics is its crystal form. It is possible to select a crystal form

that contains many defects rather than a perfect crystal. The advantage of the former form is better dissolution rate but the disadvantage is reduced stability. In a scaling up processes there are drugs of which crystal characteristics can change, i.e. can undergo polymorphism. Different polymorphic forms have different structures in the crystalline state, they can have different solubility i.e. they can have different dissolution rate. When the pharmaceutical formulator knows there is a tendency for polymorphism, the process involved should be taken into consideration as well as the selection of excipients. It is possible, by changing the shape of the particles, to influence mixing and tabletting processes. The processes involved in the production of tablets, like drug micronisation, could also influence the dissolution rate. The determination of the drug solubility is a very important step in tablet development. In order for the drug to be active it has to dissolve in body fluids and then be absorbed. To measure the drug's intrinsic dissolution rate the drug is compressed close to zero porosity in a special die. The die is immersed in solvent and the drug concentration as a function of time is measured. This measurement can estimate if dissolution is likely to be a problem and the appropriate excipient(s) required, e.g. wetting agents to help overcome a dissolution problem. Before adding any excipient to the drug it is common practice to test how the drug interacts with the excipient. Placing them together in an incubator with elevated temperature and humidity for accelerated compatibility study does this.

There are a variety of purposes in adding diluents. For example the drug dose could be small ~5 mg. It is not easy to handle such small tablets so it is necessary to add material that will increase the tablet volume. The appropriate choice of diluent for the formulation must be compatible with the other tablet ingredients. For example calcium salts are not suitable diluents for tetracyclines since they interfere with the absorption. Another interaction is between amine bases or salts with lactose or alkaline lubricant. This latter interaction results in tablet discoloration. Lactose is one of the popular diluents. It is important to note that different types of lactose have different properties. Carbohydrates are also popular diluents since they are not toxic, have reasonable taste, reasonable dissolution profile and the most importantly they enhance tablet cohesiveness (so act as binders). Excipients that are water-soluble and particularly diluents (since they constitute a large proportion of the tablet) are an important factor for good dissolution. Water content is an important diluent parameter that can influence drug stability, e.g. in preparing Aspirin® tablets it is best that the diluent contains no moisture since acetyl-salicylic acid reacts with water. Two other criteria for diluent selection are the production process and diluent

strength characteristics. The latter criterion is illustrated in plots of crushing strength as a function of applied force.

Binders are used for adding cohesiveness to a tablet, so the powder upon compression would became a strong compact. It is important that the tablet is not too strong since this can influence drug dissolution from tablet. The primary criterion for choosing a binder is its inertness with the other tablet ingredients (compatibility). Secondly, the cohesiveness ability since without enough cohesiveness a tablet would not be formed. A third reason is the ease of binder use in the production process.

The purpose of a disintegrant is to facilitate tablet breakage into smaller particles upon contact with body fluids. There are six basic groups of disintegrants: starches, clays, celluloses, algins, gums and miscellaneous. Disintegrants work by several mechanisms. The first group is of disintegrants that propagate capillary effects (water uptake by capillary forces), materials that belong to the group are starches and some forms of microcrystalline cellulose like Avicel. The second group is of disintegrants that swell. The problem is that some of the excipients belonging to the second group form a wet mass causing tablet swelling (e.g. powdered gum like tragacanth), but since it is also good binder, a concentration above 5% is not recommended. The third mechanism belongs to the gas producing disintegrants. These are good when fast disintegration is needed. In the production process involving this third type of disintegrants, it is necessary that in the tabletting room the relative humidity is low. In this process it is also necessary to add the disintegrant just before tablet compression to avoid moisture uptake by them. Their composition is similar to that of effervescent tablets. The most common is a mixture of citric acid with tartaric acid with carbonates or bicarbonates.

There are also two other types of disintegrant mechanism that are not so popular. The first is a disintegrant of the enzyme type. They work by acting on specific binders that are used in wet granulation process. The reaction between the enzyme and the binder in the presence of body fluids generates the tablet disintegration reaction. The second type of disintegrants is the melter disintegrants. The tablet ingredients are 'locked' inside these disintegrants. Since they have low melting points, at body temperature they become liquid and the contents are poured out inside the body.

It is recommended that the disintegrant is both inside the granules and outside it, i.e. intergranular and extragranular. As the portion of disintegrant increases in the intragranular part relative to its extragranular part the tablet becomes harder. Addition to the extragranular part is good for fast disintegration rate whereas adding to the intragranular part is important for breaking each granule into fine particles.

The next excipient that will be discussed is the lubricant. Most of the lubricants are hydrophobic so it is important to put disintegrants that are hydrophilic like starch in the extragranular part. Combinations of disintegrant and lubricant that are added before production are called running powder (Lieberman et al., 1989). This simultaneous addition prevents a possible pitfall, since the lubricant could coat the disintegrant preventing water from penetrating thus reducing the effectiveness of the disintegrant.

The lubricant's role is to reduce friction between the powder and the die wall. Lubricants enable the transmission of tablet machine compression force and also help reduce the ejection force. There are two types of lubricants, the fluid and the boundary type. The first type acts by the creation of separation between the two surfaces that move past one another, i.e. between the metal surface of the machine and the powder surface. Mineral oil is an example of this type of lubricant. The second type act by adhering to the tablet press metal-surfaces in the die using the polar part of the molecule, e.g. magnesium stearate. The first type is considered inferior relative to the second one since without the polar portion its adherence to the metal surfaces of the die wall is poor. As a guideline, lubricant mixing time and lubricant concentration should be the minimum possible. This is because as the amount of lubricant inside the tablet increases there is more chance the lubricant would weaken the tablet or impair the dissolution process (most lubricants being hydrophobic). In the production process it is worth monitoring carefully the mixing time, mixing speed and batch size since they all influence the lubricant's performance. With respect to tablet dissolution it is preferable use soluble lubricants but their lubrication efficiency is less, relative to the hydrophobic ones. Sodium lauryl sulphate is an example of a soluble lubricant. Since it is a weak lubricant it must be added in higher concentrations than the hydrophobic ones. Sometimes, combinations between hydrophilic and hydrophobic lubricant give the best balance with respect to lubricity, tablet hardness and disintegration time, e.g. starch is added to the lubricant in a range of 1:1-1:4 (Lieberman et al., 1989).

Glidants are used to enhance the powder flow and also help in the particle arrangement inside the die before the compression stage. In addition, glidants help to prevent segregation due to excess vibration. They act by reducing interparticulate friction. Uniformity of weight property is directly related to the consistency of flow since the same amount of powder should enter the die for each tablet. If upon addition of glidant the flow is still poor, it is possible to force feed the powder into the tabletting machine. Corn starch is an example of a glidant that changes flow properties with changes in its concentration. At low concentration it is used as glidant, at concentrations above 10% it does the opposite by reducing powder flow. Another characteristic it has is that it also acts as a disintegrant. As mentioned in relation to lubricants, high concentrations are not recommended, e.g. a high concentration of a hydrophobic glidant like talc has a bad influence on the dissolution rate.

Talc, starch and most of the lubricants belong to the antiadherents group. In certain cases although there is a lubricant in the formulation it is necessary to add materials from this group. This is because sometimes the powder adheres to the punches or to the die wall, that can be seen in a phenomenon termed 'picking' in the tablet, giving the tablet rough surfaces. In such a case materials of this group should be added or there should be an increase in lubricant concentration if it belongs to this group. Formulations containing high concentrations of vitamin E need antiadherent since they have a tendency to 'picking' (Lieberman et al., 1989). Usually, antiadherents are hydrophobic but there are also water soluble ones like DL-Leucine that are suitable for tablets prone to 'picking' and dissolution problems.

After the formulation ingredients are chosen, and the powder is ready for compaction it is important to test powder flow properties. As mentioned earlier the powder flow is very important for the uniformity of weight parameter and this problem is very relevant to today's world because very fast tabletting machines are in use today. Usually pharmaceutical powders are cohesive, so they tend to move together as one mass like flour or wet sand. The aim is that they will flow as freely as dry sand. There are a number of factors that affect powder flow and cohesion like particle size, shape and density as well as interparticulate forces, moisture and temperature. There are a number of tests that are used for prediction of powder flow. It is possible to measure angle of repose dynamically or in a static manner. In a test of angle of repose (static) there is a measured angle when a cone of powder is poured onto a flat surface. In an angle of spatula test the angle measured is the

one formed when material is raised from a flat surface out of a bulk pile. Another test of powder flow is flow through a circular orifice. In this test the rate of powder flow through the orifice is measured. Bulk density measurements could give estimates of flow of material. One of these measurements is Carr's compressibility index. In this test the powder is poured into a vessel and bulk density is calculated by measuring the powder's volume and weight. After standardizing the tapping procedure (Lachman et al., 1986), which can be done using a 'Jolting Meter', the tap density is calculated. From the latter two calculations Carr's compressibility index is calculated as:

(tapped density - bulk density) / tapped density

Multiplying by 100 will give the result in percentage terms.

Hausner's ratio is calculated as:

tapped density / bulk density

And again multiplying by 100 will give the result in percent.

Now that the tablet is ready and the tablet properties meet the specifications, the most expensive and time-consuming step still remains to be done, the stability testing. A drug should be stable through all its shelf life period. The drug can decompose into inactive ingredients, active metabolites or toxic materials. In an era of very tough competition between the pharmaceutical generic companies over the one who will mimic faster drugs the patent of which has expired, it is not practical to wait three years in room temperature conditions to see if the development succeeded. It is also important for the NDA companies—just because they waited too long for long term stability studies—because another company could enter the market faster with a drug for the same indication and be the first on the market. All modern pharmaceutical companies hence do accelerated stability studies with the use of several incubators under different environmental conditions. After the experimental manufacturing of the tablets they are packaged (e.g. in blisters) and placed in appropriate incubators and are withdrawn for testing with strict regimen. The regulatory authorities usually allow manufacturers after storing drugs for 6 months at 40°C and 75% relative humidity a permitted shelf life of three years if the tests on the tablets are satisfactory. But the manufacturers must still monitor the stability of the drug through all 3 years at room temperature.

There are not many studies on the kinetics of drugs in the solid state. The kinetics of benzoic acid derivatives (belong to the domain of pure solids kinetics) was studied by

Carstensen and Musa (1972). The decomposition curve of aminobenzoic acid was found to be sigmoidal. Hence, the rate order of the decomposition changes according to the decomposition phase. In the case of aminobenzoic acid, after the liquid begins to form the reaction becomes first order one, as the stability kinetics of drug in solution. Tablets could have even more complex kinetics since there are also possible interactions with excipients in the formulation. An example of how to calculate stability of drug in solution follows. It is given here since the first order kinetics described here is also assumed in Expha calculation (explained later) regarding the stability of the drug in the granulation process.

If one measures the amount of drug in aqueous solution, in each incubator, at different time points, a plot of concentration of drug on the y-axis against incubation time (in hours) on the x-axis is obtained. For each incubation temperature a regression trend line could thus be plotted between the experimental points, this plot is shown on Figure 2.1. Obviously, as the incubation temperature increases the slope of the line is greater. The slope of the line is defined as the decomposition constant k. Now it is possible to plot an Arrhenius relationship that is log k (k as measured from Figure 2.1) on the y-axis against reciprocals of the absolute temperature (multiplied by 10^6) on the x-axis. Regression trend line is drawn between the different log k for each incubation temperature and this line is extrapolated to 25°C as in Figure 2.2 (see Martin et al., 1983 for stability issues discussed here). The extrapolation to room temperature stands on the assumption that the decomposition continues with the same kinetics, an assumption that is not always the right one.

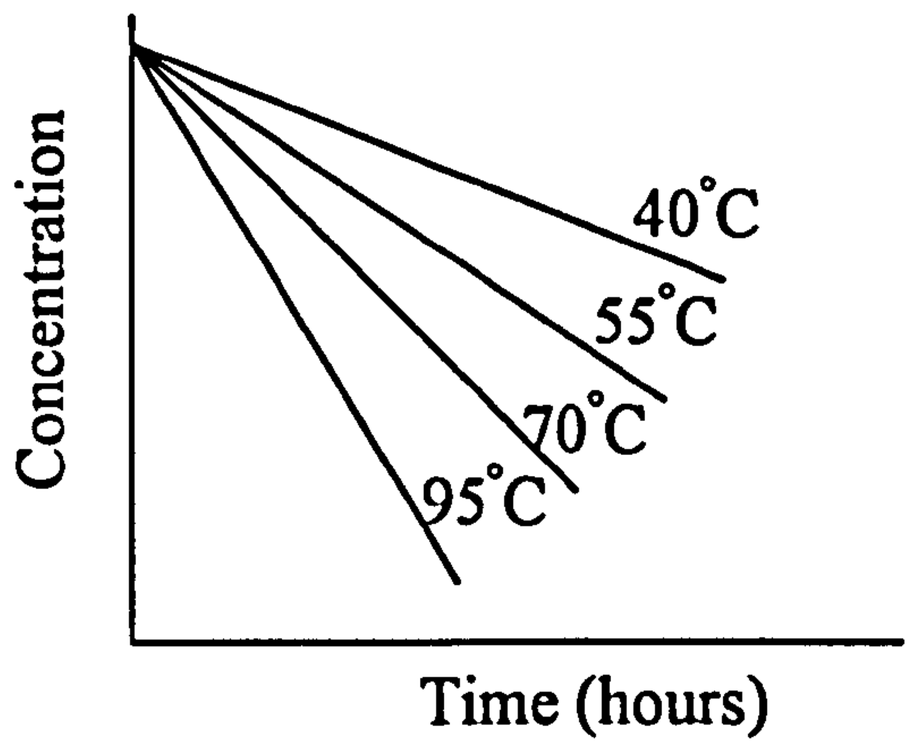


Figure 2.1: Accelerated decomposition of a drug in aqueous solution at elevated temperature.

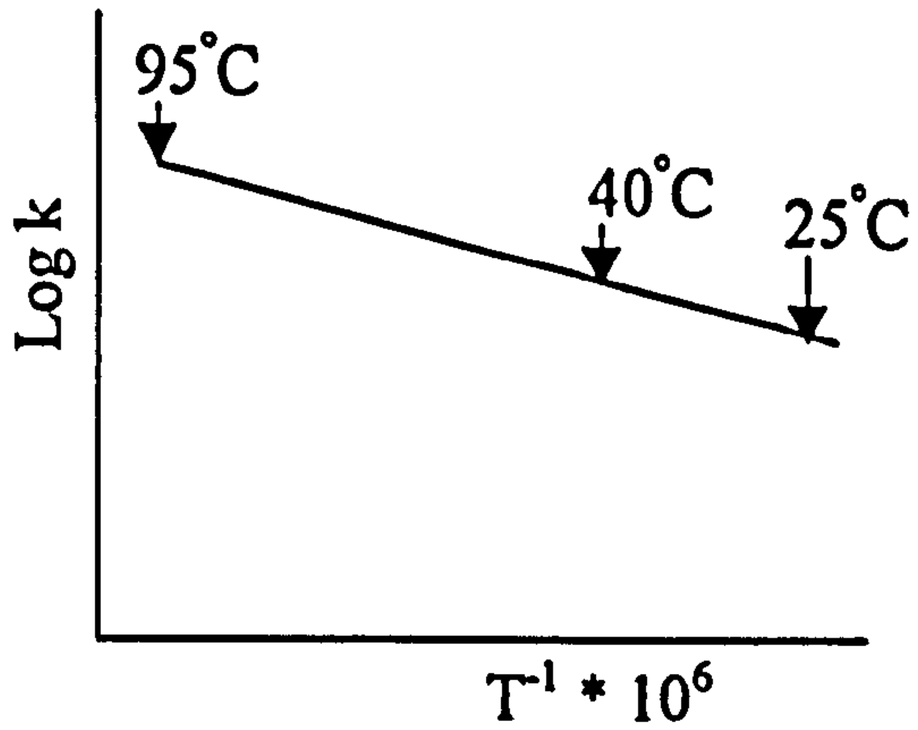


Figure 2.2: Arrhenius plot for predicting drug stability at room temperatures. T is the absolute temperature.

2.3 Regression analysis

Regression analysis is a method of finding an equation that best describes the data.

$y = a^*x + b$ is an example of a regression analysis equation. y is the measured dependent variable (response) and x is the independent variable. Residual (error) is the difference between the observed value and the predicted value by the regression line. The sum squared error (SSE) is defined in the following equation:

$SSE = \Sigma (y_{predicted} - y_{observed})^2$. The value in parenthesis is the residual. The ideal regression equation line is the one that its sum of residuals is equal to zero with the minimum SSE. From the above it is obvious why the regression line is also called the least square line. Figure 2.3 shows the change in tablet hardness as a function of compaction force. The black dots represent the experimental values whereas the line is the least square regression line.

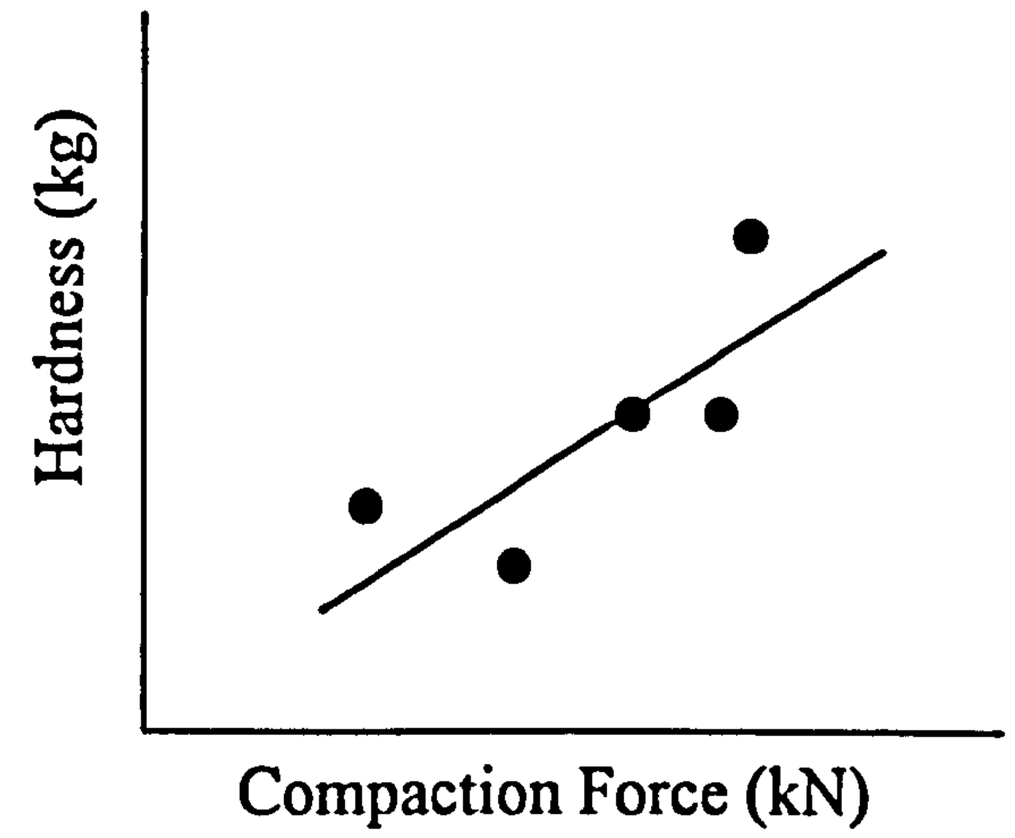


Figure 2.3: Schematic plot of tablet hardness as a function of compaction force. The line is a simple linear regression line.

It is also possible to build a complicated polynomial with several coefficients instead of the simple regression line. Figure 2.4 shows regression polynomial for the same data as in Figure 2.3. It can be seen that as opposed to the simple linear regression line presented in Figure 2.3 the regression line of Figure 2.4 fitted the data perfectly with no errors at all. The question that arises is which one of the models is better?

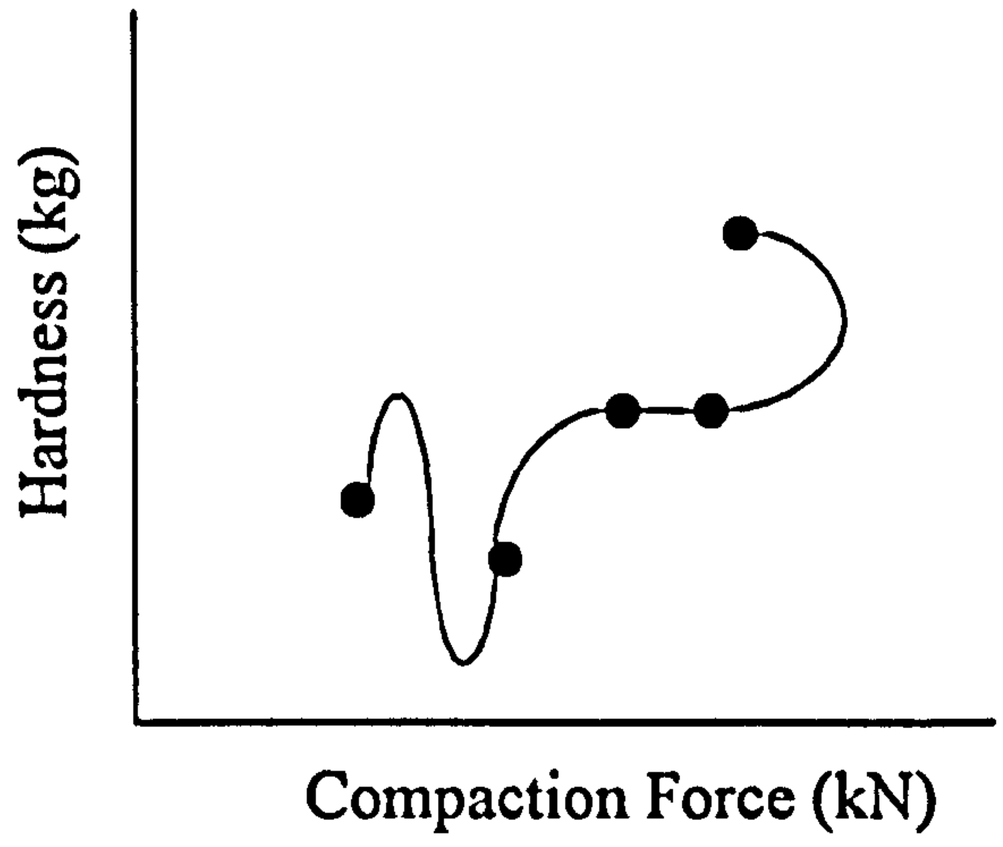


Figure 2.4: Schematic plot of tablet hardness as a function of compaction force. The line is a polynomial regression line.

It is possible to calculate a very important parameter that is called the coefficient of determination denoted by r^2. If there are several independent variables the coefficient is called the multiple coefficient of determination and is denoted by R^2. So the term multiple is added according to the number of independent variables. The multiple coefficient of determination is calculated as $R^2 = 1 - (SSE / SS)$.

SS is the sum of squares of the difference between the observed values and the mean values. The same calculation applies for the coefficient of determination (r^2). If SSE is almost equal to SS then the expression in parenthesis will approximately be equal to 1 and R^2 will approximately equal to 0. If SS is much bigger than SSE then the expression in parenthesis will approximately equal to 0 and R^2 will approximately equal to 1. What is the meaning of the R^2? If the value of R^2 is 0.9 then 90% of the sample variation in y can be explained by using the equation to predict y.

How does one know that the regression equation is good for prediction? By definition if the number of coefficients is equal to the number of cases the value of R^2 is equal to 1 and it can be stated that 100% of the sample variation can be explained by using x to predict y. But this model is usually bad for predicting. The regression line in Figure 2.3 of the simple

model of a straight line probably predicts better then the complicated model of Figure 2.4 although its R^2 is much less than the R^2 of the more complicated model. The complicated model memorises the data as opposed to the more simple equation that probably learns the data. In other words, the more complex model can predict perfectly what it was developed with but cannot predict any other data points it has not seen. Figure 2.5 shows the change in R^2 and in validation set error as a function of the number of coefficients in the regression equation. It can be seen in Figure 2.5 that as the number of coefficients goes up the R^2 goes up, this happens since the SSE goes down. When a test is applied to check the predictive ability of the regression equation (in the form of measuring the validation set error), it can be seen in the bottom plot of Figure 2.5 that from a certain number of coefficients there is a deterioration in the predictive ability. Hence, the number of coefficients should be monitored carefully. The appropriate monitoring can be done with the aid of various techniques for the selection of coefficients.

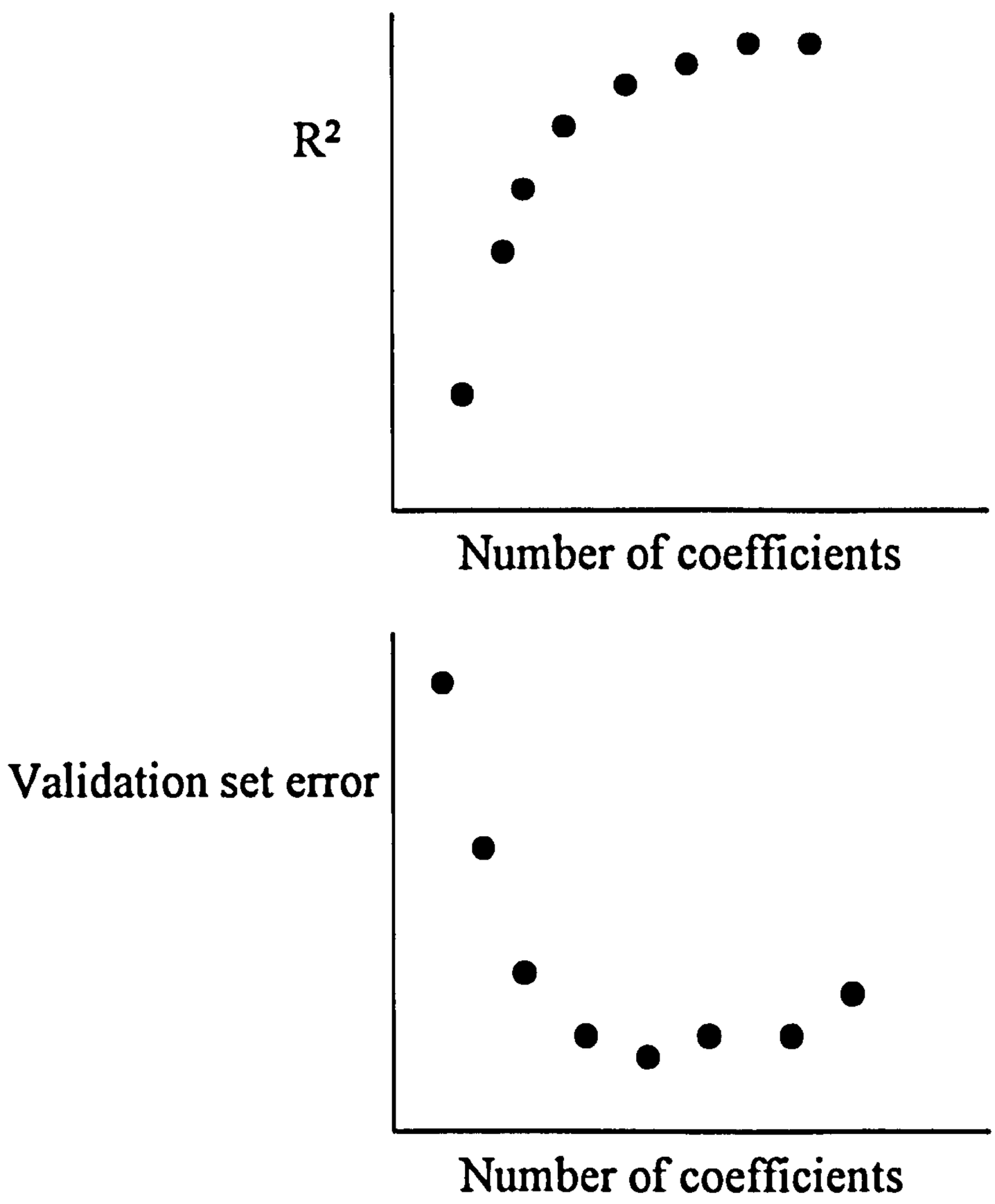


Figure 2.5: The change in R^2 (upper plot) and in validation set error (bottom plot) as a function of the number of coefficients in the regression equation.

Regression is a technique used in numerous research fields and the statistical inferences derived from this can be very interesting. Recalling the example in the introduction, a study was conducted by taking interviews with approximately 1000 women. The 2 variables measured in the study were financial reward at age 40, which was the dependent variable, and premarital pregnancy as the independent variable. The article concluded that couples beginning marriage with the bride already pregnant faced lower income and living standards and 22% fewer assets than couples with no premarital pregnancy. The paper concluded hence, a cause and effect relationship, but it may be that the socio-economic level of the women contributed to their premarital pregnancy and less money at the age of 40. The data that they collected is called observational data. In observational data it is not possible to control the x values and one cannot draw a cause and effect relationship although the statistical tests are significant. In experimental data the independent variables are set in advance and it is possible to draw a cause and effect relationship. For example, in a set of experiments only compaction force was changed and the response measured was hardness. It could be stated that change in the compaction force caused the tablet to be harder.

2.4 ANN

There are a large number of texbooks, which provide a good introduction to ANN field. (Nelson & Illingworth, 1991; Aleksander & Morton, 1992; Haykin, 1994; Beale & Jackson, 1994; Bishop, 1996). The most readable of these are those of Nelson & Illingworth and Beale & Jackson. However, Nelson & Illingworth's book is quite superficial that gives just a 'feel' to ANN, since it is a brief course not intended for people with a scientific background. Beale & Jackson's book gives a more robust foundation to ANN without delving into complicated mathematics. This introduction to ANN is based in part on these books. The information here is also based on Hines (1997), Hagan et al. (1996) and Demuth & Beale (1998). The book by Gallant (1993) describes an expert system that incorporates ANN, and this concept of incorporating ANN into expert systems was also used in this study.

An ANN is a mathematical model that uses as its basic component the model of the neuron in the brain. This basic unit in ANN is called a perceptron (Hagan et al., 1996). The neuron in the brain receives its inputs from other neurons and so does the perceptron. The strength

of the input from the other neurons depends on the synaptic strength. In ANN this synaptic strength is called the weight of the connection. On receiving enough stimulation, the neuron in the brain 'fires' and feeds its output to other neurons that are connected to it. Similarly, the perceptron receives a signal and according to its activation function sends its output to other connecting neurons.

Figure 2.6 shows ANN with a single neuron in action. The goal is to train ANN so that when input "x_1" is presented it will generate output of "$y_{1\ obs}$" and when input "x_2" is presented it will generate output of "$y_{2\ obs}$". In Figure 2.6 there is ANN with just one neuron. The value of the single input unit is multiplied by the weight denoted by the letter, "W". To allow more flexibility in the system a bias value is introduced denoted by the letter "B". The bias summed with the value of the input unit multiplied by "W" produces the calculated "X" value. The bias is always multiplied by 1 and generally there is a bias in ANN. The bias can be considered just as having another weight attached to an input unit with a constant value of 1. So the ANN could be seen as one with 2 input units and 2 weights. The calculated "X" value enters into the activation function of the perceptron which outputs the calculated "$y_{1\ cal}$" value. In the next iteration "x_2" is the input value and the output of the ANN is "$y_{2\ cal}$".

"$y_{1\ obs}$" and "$y_{2\ obs}$" are the desired outputs to the inputs of "x_1" and "x_2" respectively. The difference between the observed and the desired output of the ANN is the residual. The Sum Squared Error (SSE) is the sum of the squared residuals. The SSE in Figure 2.6 is defined as:

$SSE = (y_{1\ obs} - y_{1\ cal})^2 + (y_{2\ obs} - y_{2\ cal})^2$.

By adjusting the weights ANN can minimise the SSE. Looking at the ANN from Figure 2.6 there are two weight variables to be optimised: "B" and "W", the optimisation process aim to find the values of these variables that would give the lowest error (minimum SSE). The process of minimisation of the SSE by the ANN is called the learning/training process. A learning rule (also known as training alogorithm) is a procedure for modifying the weights and biases of a network. Backpropagation is a training algorithm that is commonly used to adjust the weights in a multilayer network. In this learning method the weights adjustments are propagated backwards from the output neurons to the input neurons.

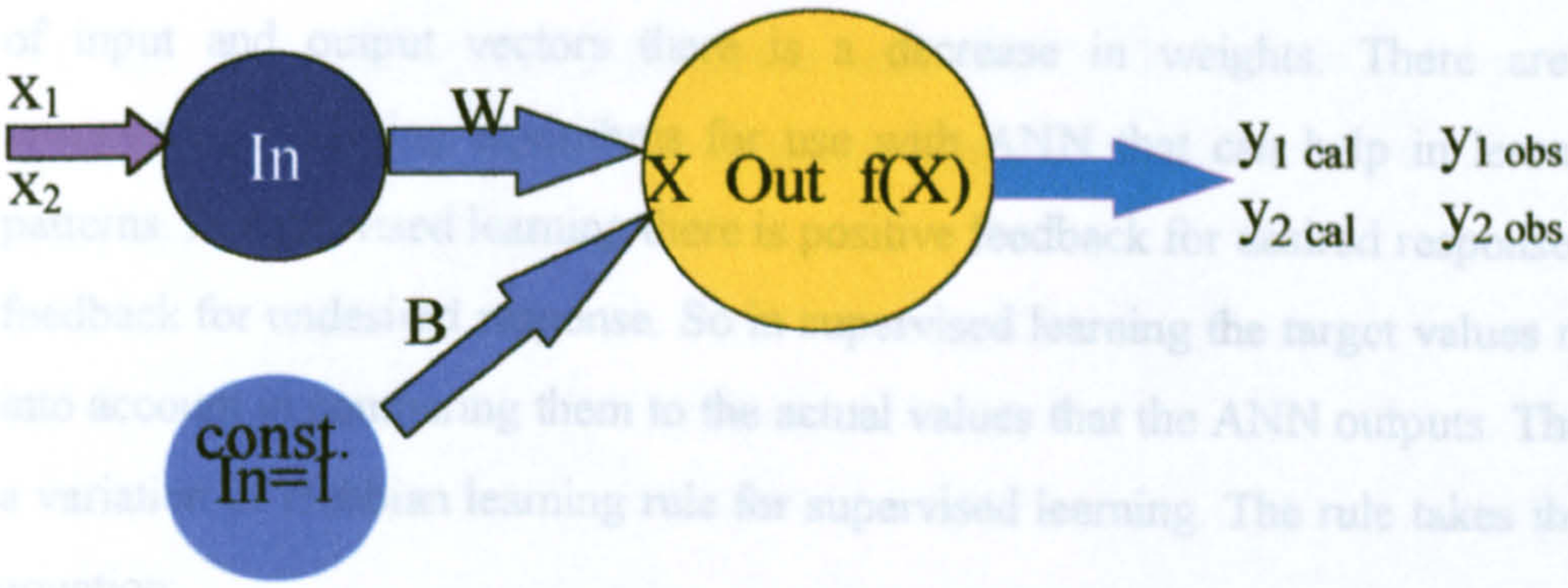


Figure 2.6: Inside ANN of a single neuron. "x_1" and "x_2" are the input values. Abbreviations: "In" is input unit and "out" is output perceptron. The sum $x_1 \cdot W+B$ (= X) is entered to the perceptron into its activation function "f(X)" and the output is "$y_{1\ cal}$". The required output is "$y_{1\ obs}$". The same calculation is repeated on "x_2". In the training process "W" (weight) and "B" (Bias) are adjusted in order to minimise the SSE.

2.4.1 Learning by backpropagation

This section explains the backpropagation algorithm for ANN training. Backpropagation is one form of supervised learning as opposed to unsupervised learning. Unsupervised learning does not require any information regarding the target output. The unsupervised Hebb rule is demonstrated in the following equation:

$$\mathbf{W}(\text{new}) = \mathbf{W}(\text{old}) + \alpha\, \mathbf{a}_q\, \mathbf{p}_q^{\,T}$$

Where **W**(new) are the new weights. **W**(old) are the old weights to be updated. α is the learning rate and $\mathbf{a}_q$, $\mathbf{p}_q$ are output and input pair for a given q data point respectively. The superscript T is denoted for transposition of the input vectors from rows to columns to enable multiplication with output vector columns. It can be seen from the equation that when the input is positive and the output of the ANN is positive the new weights are bigger. The change in weights is proportional to the size of input and output vectors. In the brain when one neuron excites an adjacent one (input to the neuron) the efficiency of

excitation with more and more stimulus is increased, this is done by a physical mechanism between the cells in the brain. There is evidence that some brain cells do behave according to this Hebbian learning pattern. It can also seen from the equation that if both input and output pair are negative there is a proportional increase in the weights. For opposite signs of input and output vectors there is a decrease in weights. There are many other unsupervised learning algorithms for use with ANN that can help in learning the input patterns. In supervised learning there is positive feedback for desired response and negative feedback for undesired response. So in supervised learning the target values must be taken into account in comparing them to the actual values that the ANN outputs. The delta rule is a variation of Hebbian learning rule for supervised learning. The rule takes the form of the equation:

$$\mathbf{W(new) = W(old) + \alpha\,(t_q - a_q)\,p_q^T}$$

Where all the parameters are like the previous equation except the $\mathbf{t_q}$ which is the relevant target vector of q data point. The delta rule got its name from the difference between desired and actual output. It is also called Widrow-Hoff algorithm (Widrow & Hoff, 1960) after the researchers who developed it. The delta rule is used to minimise the SSE and is precursor to the backpropagation algorithm which will be presented now.

Like the Widrow-Hoff algorithm the backpropagation algorithm also aims to minimise the SSE. The function that the training process tries to minimise is called the performance function. Looking at the ANN from the previous example there are 2 variables to be optimised the bias B and the weight W. The aim is to find the values of these variables that will give the lowest error (minimum SSE). Suppose one wants to plot the error function of the surface. First put values of 4, -4 for both W and B. The ANN is trained with these values and the SSE for data point (4, -4) is calculated (SSE = 0.2, see Figure 2.7). Then at specific interval the next point is selected and using the ANN its SSE is calculated. In the same way thousands of points can be calculated by taking different values of W and B and the result is shown in Figure 2.7. On the right there is a contour plot of the error surface and on the left is the error surface. The height of the surface is the SSE. The aim is to get to the lowest SSE. At the point of lowest SSE the weight and bias are at their appropriate values and the ANN has learned. Backpropagation tries to minimise the error by gradient descent. A simple analogy is a ball with no inertia that represents the network rolling

around the error surface. The ball always rolls in the steepest direction until it stops at the bottom of a valley, which is the error minimum. The network path (called trajectory) crosses the contour lines at approximately 90 degrees in a method called steepest descent. As an example of minimisation path, one can look at Figure 2.7 and see the starting guess (-2, -2). The optimisation path moves according to the law of gravity, in the manner described earlier till it reaches the minimum point. It stopped there since it cannot go down from this point.

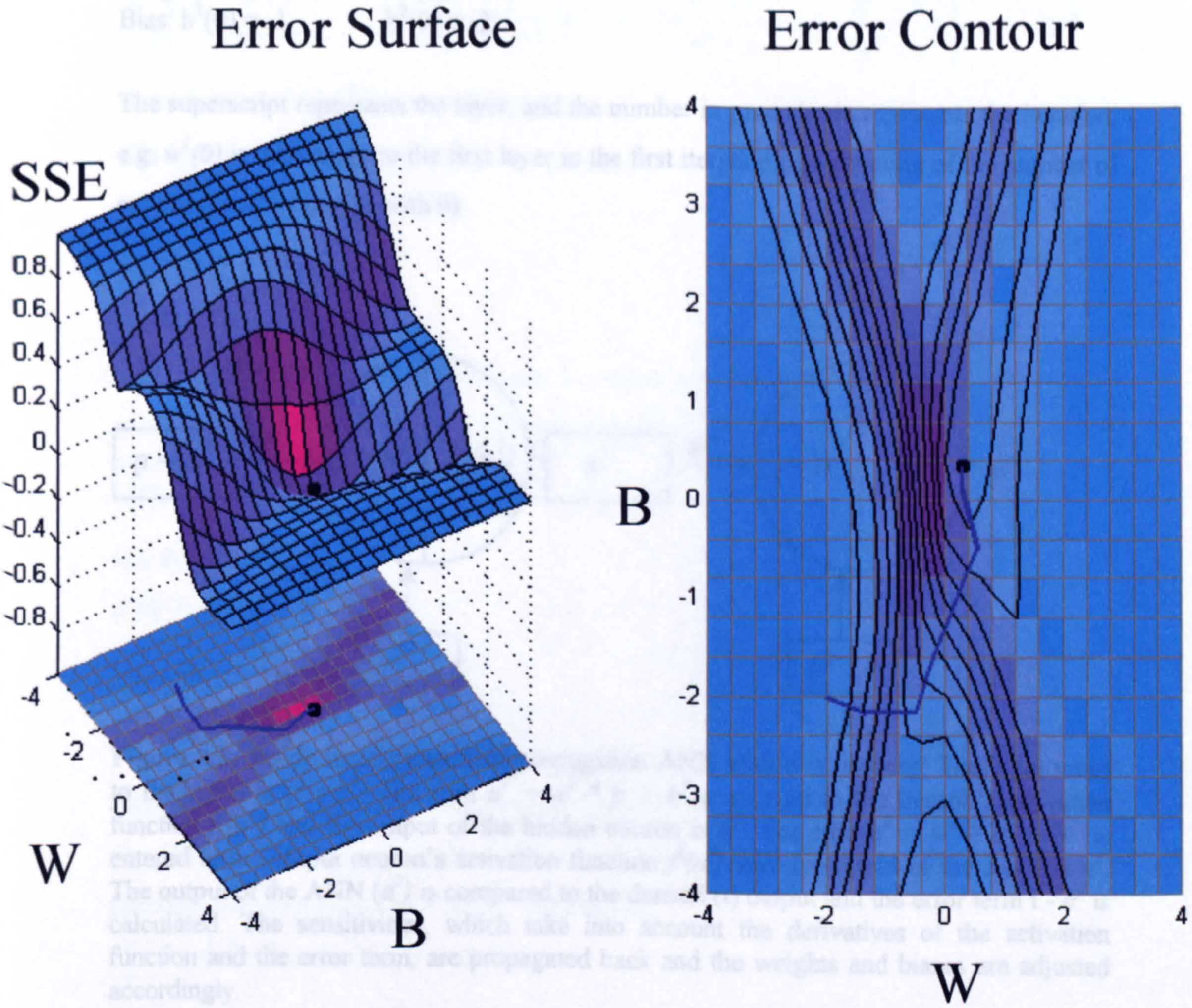


Figure 2.7: Minimisation pathway starting from the point -2, -2. The weight (W) and bias (B) are changed at each iteration until the minimisation leads to the minimum point.

Figure 2.8 illustrates a multilayer ANN with one hidden neuron and one output neuron. This ANN will be used to demonstrate how ANN trains in a backpropagation. The input to the network is p and the goal is to train the ANN to generate output of t when presented with this input. The input to the hidden neuron in the first layer can also be seen as the output from layer 0 and be designated as a^0. Here are the ANN parameters to be trained on:

Input: $p = a^0 = 0.5$

Target: $t = 0.9$

The ANN weights and biases are:

Weights: $w^1(0) = 1$ $w^2(0) = 2$
Bias: $b^1(0) = -1$ $b^2(0) = -2$

The superscript represents the layer, and the number in parenthesis represents the iteration, e.g. $w^1(0)$ is the weight to the first layer in the first iteration (the counting of the number of epochs for ANN begins with 0)

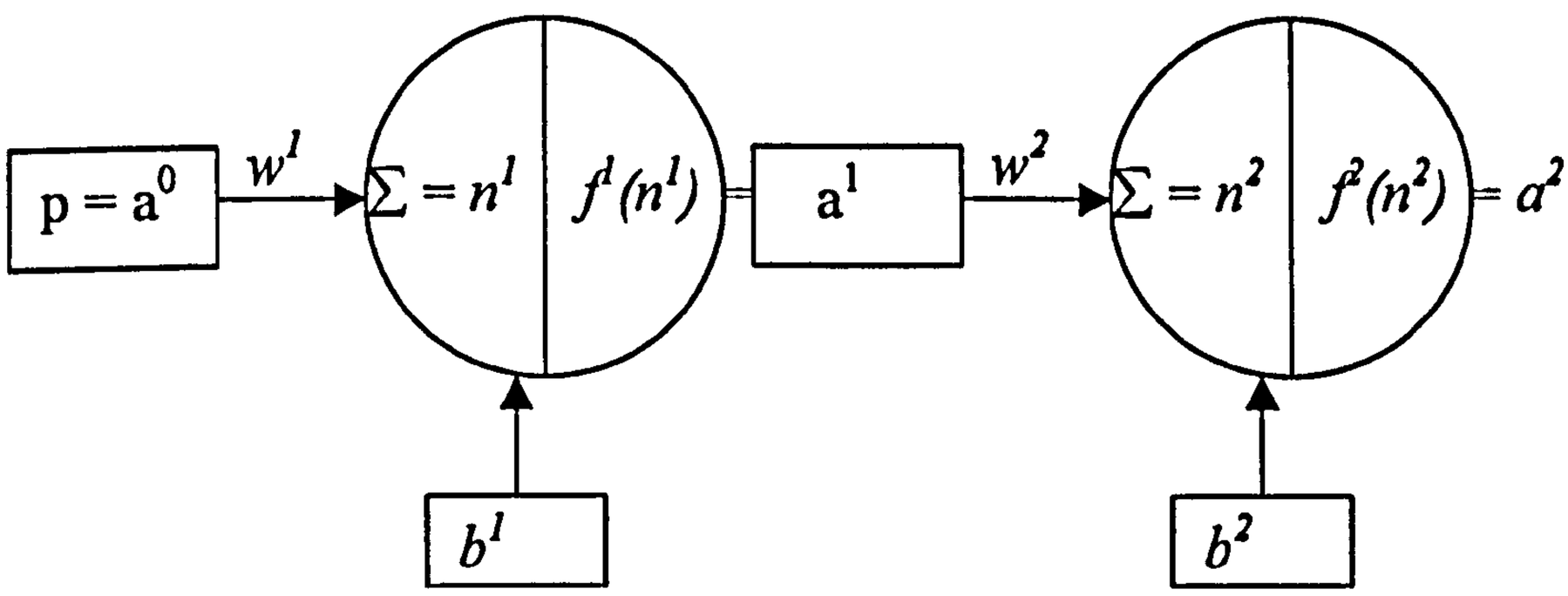


Figure 2.8: Inside feedforward backpropagation ANN with two neurons. The input value to the ANN is $p = a^0$. The sum $n^1 = w^1 * p + b^1$ is entered to the neuron's activation function $f^1(n^1)$ and the output of the hidden neuron is a^1. The sum $n^2 = w^2 * a^2 + b^2$ is entered to the output neuron's activation function $f^2(n^2)$ and the output of the ANN is a^2. The output of the ANN (a^2) is compared to the desired (t) output and the error term $t - a^2$ is calculated. The sensitivities, which take into account the derivatives of the activation function and the error term, are propagated back and the weights and biases are adjusted accordingly.

The parameters so far were important for calculating the forward pass of the ANN. The activation functions are important for calculating the forward pass and their derivatives are important for the backward steps of adjusting the weights. The learning rate is an important parameter of the backward steps. Here are additional parameters that are important for the learning process of the ANN:

Learning rate: $\alpha = 0.1$

Activation function of the first layer:

$$f^1(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$

Its derivative is:

$$\frac{df^1(n)}{dn} = \frac{d}{dn}\left(\frac{e^n - e^{-n}}{e^n + e^{-n}}\right) = -\frac{e^n - e^{-n}}{\left(e^n + e^{-n}\right)^2}\left(e^n - e^{-n}\right) + \frac{e^n + e^{-n}}{e^n + e^{-n}} = 1 - \frac{\left(e^n - e^{-n}\right)^2}{\left(e^n + e^{-n}\right)^2} = 1 - \left(a^1\right)^2.$$

Activation function of the second layer:

$$f^2(n) = n$$

Its derivative is:

$$\frac{df^1(n)}{dn} = \frac{d}{dn}(n) = 1$$

Calculation of the forward pass to generate the ANN output:

$n^1(0) = w^1(0)\ p + b^1(0) = (1)\ (0.5) + (-1) = -0.5$
$a^1(0) = f^1(n^1) = -0.46212$
$n^2(0) = w^2(0)\ a^1(0) + b^2(0) = (2)(-0.5) + (-2) = -3$

and the output of the ANN is:

$a^2(0) = f^2(n^2) = -3$

The error term of the network is:

$e = (t - a^2) = (0.9 - (-3)) = 3.9$

Calculation of the new weights and biases of the ANN:

For abbreviation, the calculation of the derivatives for the first and second activation functions will be denoted as F^1 and F^2 respectively.

$F^1 = 1-(a^1)^2 = 1 - (-0.46212)^2 = 0.78645$
$F^2 = 1$

In backpropagation the sensitivities (denoted by s) are propagated back:

$s^2 = -2\ F^2 e = (-2)(1)(3.9) = -7.8$
$s^1 = F^1 w^2 s^2 = (0.78645)(2)(-7.8) = -12.269$

Now that the sensitivities for the first and second layer were calculated the weights and biases can be updated:

$w^2(1) = w^2(0) - \alpha s^2 a^1 = 2 - (0.1)(-7.8)(-0.46212) = 1.6395$
$w^1(1) = w^1(0) - \alpha s^1 a^0 = 1 - (0.1)(-12.269)(0.5) = 1.6135$
$b^2(1) = b^2(0) - \alpha s^2 = -2 - (0.1)(-7.8) = -1.22$
$b^1(1) = (0) - \alpha s^2 = -1 - (0.1)(-12.269) = 0.2269$

In more complex ANN topologies the calculations of the forward and backward backpropagation use matrix algebra, but the calculations are in exactly the same manner. The backpropagation calculations will henceforth be presented in a general form using

matrices and vectors. Small bold letters denote vectors and bold capital letters denote matrices:

Forward Propagation

$\mathbf{a}^0 = \mathbf{p}$

$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1})$ for $m = 0, 2, \ldots, M-1$

Where m is the layer number and M represent the output layer.

The calculated output of the ANN is $\mathbf{a} = \mathbf{a}^M$.

Backward Propagation

The sensitivities are propagated back and they take into account the derivatives **F(n)** and the error term **(t-a)**.

$\mathbf{s}^M = -2\mathbf{F}^M(\mathbf{n}^M)(\mathbf{t}-\mathbf{a})$

$\mathbf{s}^m = \mathbf{F}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}$ for $m = M-1, \ldots, 2, 1$

Where the superscript T in $(\mathbf{W}^{m+1})^T$ represent the transposition of the matrix such that the columns of $\mathbf{W}^{m+1}$ become rows in $(\mathbf{W}^{m+1})^T$ and the rows in $\mathbf{W}^{m+1}$ become columns in $(\mathbf{W}^{m+1})^T$. This operation is important for the matrix multiplication.

Weight & bias update

Finally the weights and biases are adjusted with α learning rate.

$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha\, \mathbf{s}^m (\mathbf{a}^{m-1})^T$

$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha\, \mathbf{s}^m$

2.4.2 Heuristic variations on backpropagation: momentum and adaptive learning rate.

The adding of momentum and/or adaptive learning rate can improve performance and convergence speed of backpropagation. Momentum helps in overcoming barriers by jumping over local minima. It causes the algorithm to go faster in the trajectory direction. Momentum acts to screen out the noise and with only the general/average trend left, it is like a filter. Due to the filtering property of momentum it can help ANN to relate not just to the current gradient but also to recent trends in the error surface. As the momentum term gets larger it filters more noise. As the trajectory becomes smoother the weight changes are less drastic and fewer oscillations can be detected in a plot of SSE versus the number of iterations. The implementation of the momentum term is done at the stage of updating the

weights and biases. Adding the momentum term changes the equations for updating the weights and biases to:

$$\Delta W^m(k+1) = \gamma \Delta W^m(k) - (1-\gamma)\,\alpha\, s^m (a^{m-1})^T$$

$$\Delta b^m(k+1) = \gamma \Delta b^m(k) - (1-\gamma)\,\alpha\, s^m$$

where γ is the momentum term.

The momentum term takes values between 0 and 1. If the momentum takes the value of 1 the right side of the equations above became 0, and there is no change in the weights and biases. If the momentum takes the value of 0 there is no momentum and the weights and biases are updated as in regular backpropagation.

Figure 2.9 shows a more intuitive explanation of momentum in backpropagation. The demonstration of the backpropagation path is of ANN with only one input unit, one neuron and one weight. On the x-axis is the weight of the neuron and on the y-axis is the SSE. The ball always rolls in the steepest direction (the derivative of the activation function demonstrated earlier), according to the law of gravitation, until it stops at the bottom of a valley; this is the error minimum. In order to avoid getting stuck in local minimum the momentum term is added and if large enough the ball jumps over the local minimum to fall into global minimum, hence reaching the lowest error. If the momentum is too low it will not jump over the local minimum. If it is too high it will jump over the global minimum.

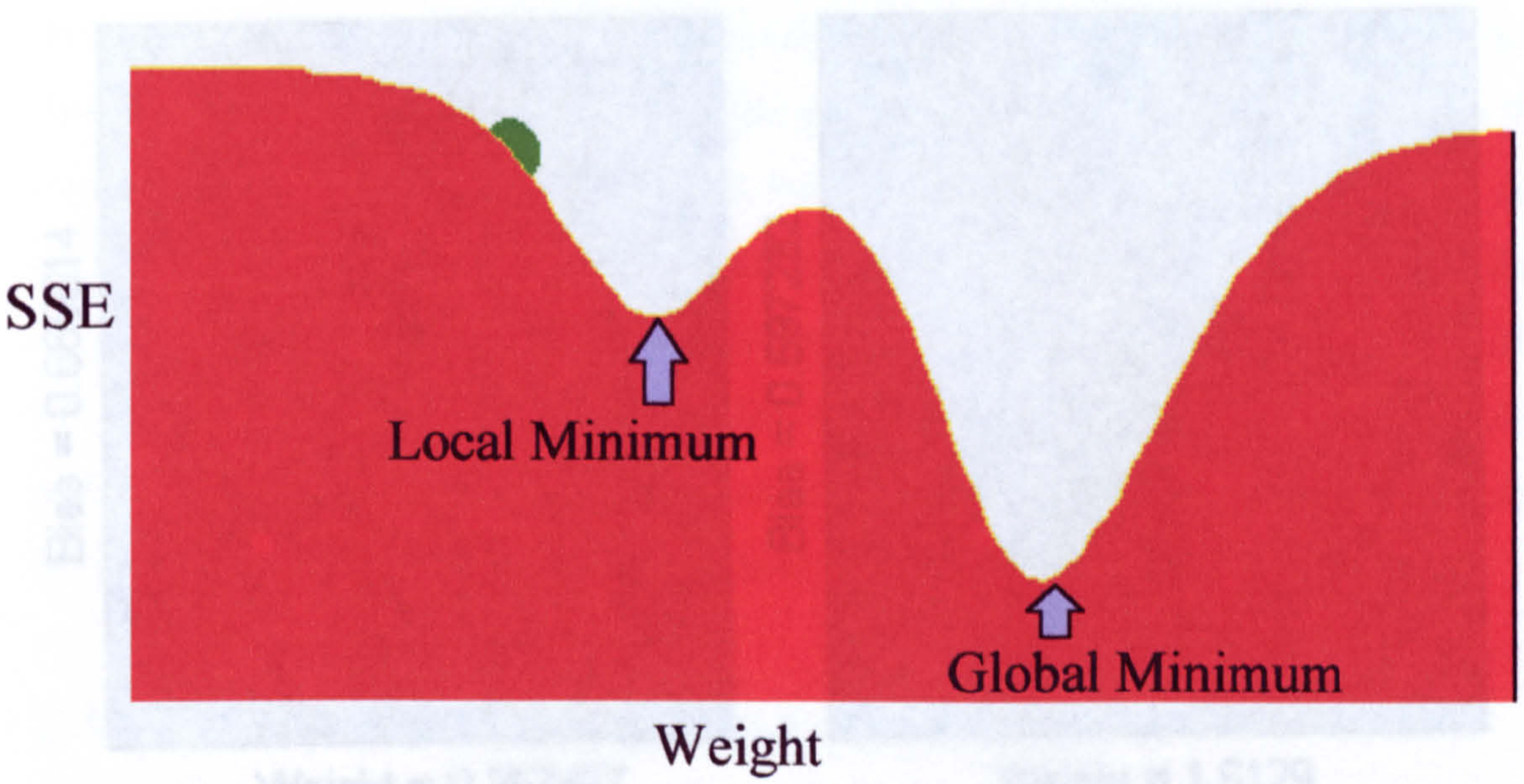


Figure 2.9: The green ball will fall, according to the gravitation laws, to the local minimum but with the aid of the right amount of momentum it could reach the global minimum.

Learning rate is the size of the steps that an ANN takes in the error surface. If it is too small it will take a lot of steps for the ANN to arrive at the error minimum. The learning rate required is the one that produces large steps so the ANN learns quickly, and yet small enough to produce short steps when needed. The solution to the problem is a variable learning rate. A variable learning rate is used when there is a need that the size of the steps will take into account the error surface. When the surface is flat it is preferable the step size is large. When the surface is steep there is preference for a small learning rate. Figure 2.10 illustrates the backpropagation path of ANN with single neuron with one weight and one bias. On the left is an illustration of adequate learning rate and on the right an illustration of a learning rate that is too large. It can be seen that the large step does not allow the ANN to settle at the error minimum because each step jumps over the error minimum.

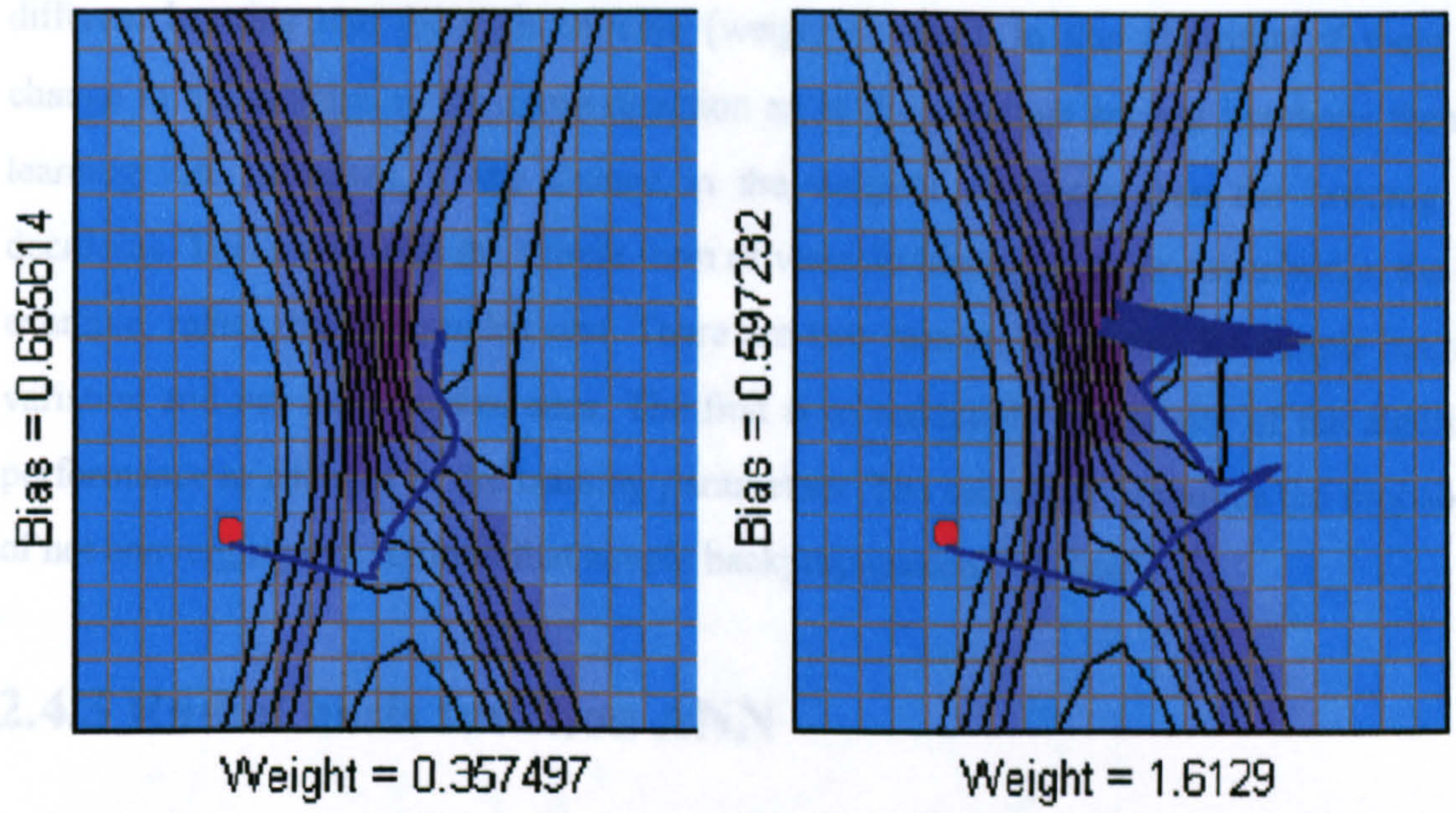


Figure 2.10: On the left picture is ANN with adequate learning rate and on the right is ANN with learning rate too large.

The two questions to ask therefore are:

How can one know the error surface?

How much change can there be in the step size?

There is a common algorithm for variable learning rate (Hagan et al., 1996) that has 3 rules:

1. When the SSE increases below a specified value of ζ (usually one to five percent) then the weight update is accepted and the momentum (if used) and adaptive learning rate are unchanged.

2. When the SSE increases by more than ζ then the new weights are discarded and the learning rate is multiplied by value $0<\rho<1$ and the momentum term is set to 0.

3. When there is a decrease in the SSE then the new weights are accepted. The learning rate is multiplied by $\eta>1$. If the momentum (γ) was previously set to zero, it is now set to its original γ value.

So the learning rate is dependent upon 3 new parameters ζ, ρ, and η. If momentum as well as adaptive learning rate is in use then the algorithm is dependent upon 5 parameters, whereas simple backpropagation was dependent on just one parameter of α, the value of

the learning rate. There are numerous algorithms of variable learning rate, e.g. one that has different learning rate for each variable (weight or bias). In this algorithm if there is a change in the weights in the same direction as in the previous several iterations then the learning rate increases, if the change in the weights alternates then the learning rate decreases. This study uses the simple form of variable learning rate as described in the first example, rather than a complex one. There are two reasons for doing the simple heuristic variation and not the complex ones. The first is to reduce the sensitivity of the algorithm performance to changes in the learning parameters. The second is to reduce the probability of not converging to a solution that simple backpropagation will find.

2.4.3 Radial basis function ANN

Radial basis function (RBF) ANN is different from ANN of the usual multilayer perceptron (MLP) type. This section will explain how they differ and how they work. The output layer neurons of RBF ANN performs a linear transformation in the activation function, but their weights are usually adjusted according to least-square algorithm and not with backpropagation as in MLP. The hidden layer neurons are radial basis function neurons. Each one of them does local mappings to the inputs. So, RBF ANN does local mapping whereas MLP does global mapping to the inputs. Figure 2.11 shows how they map differentially their input data.

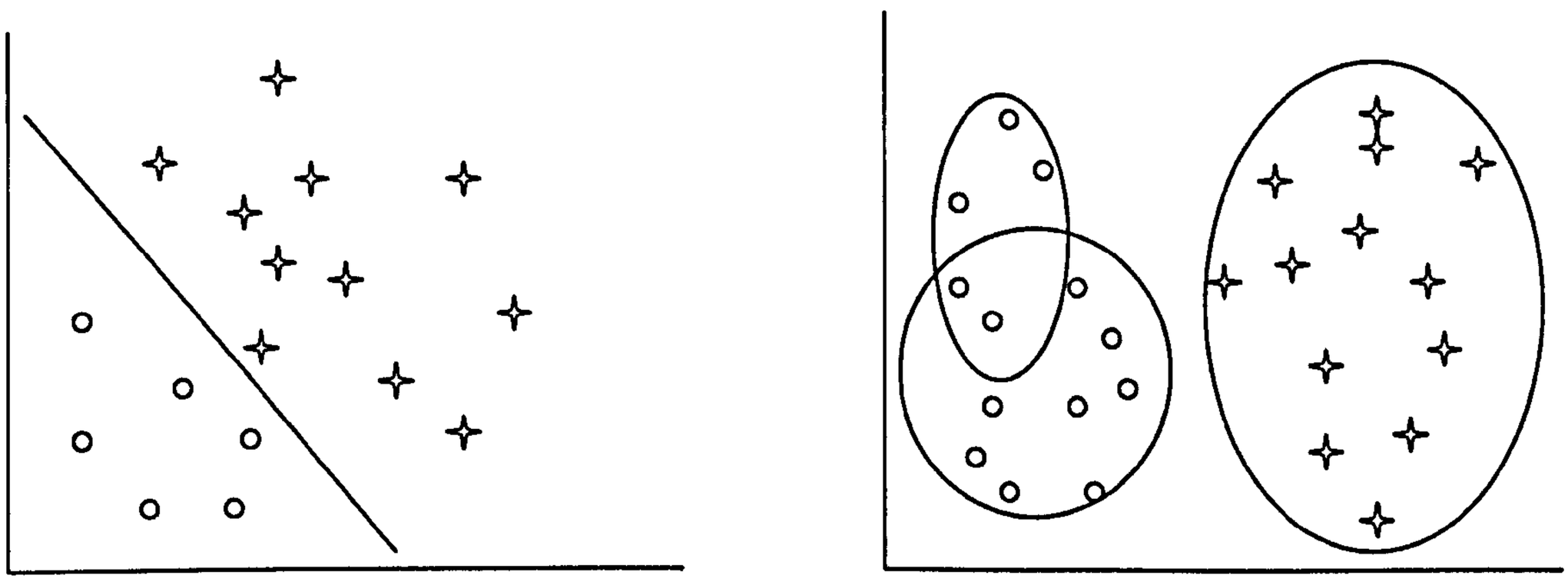


Figure 2.11: Multilayer perceptron (MLP) global mapping on the left figure, versus radial basis function (RBF) ANN local mapping on the right figure.

Each RBF neuron has its own receptive field. Figure 2.12 shows the receptive fields in the input space of each RBF neuron.

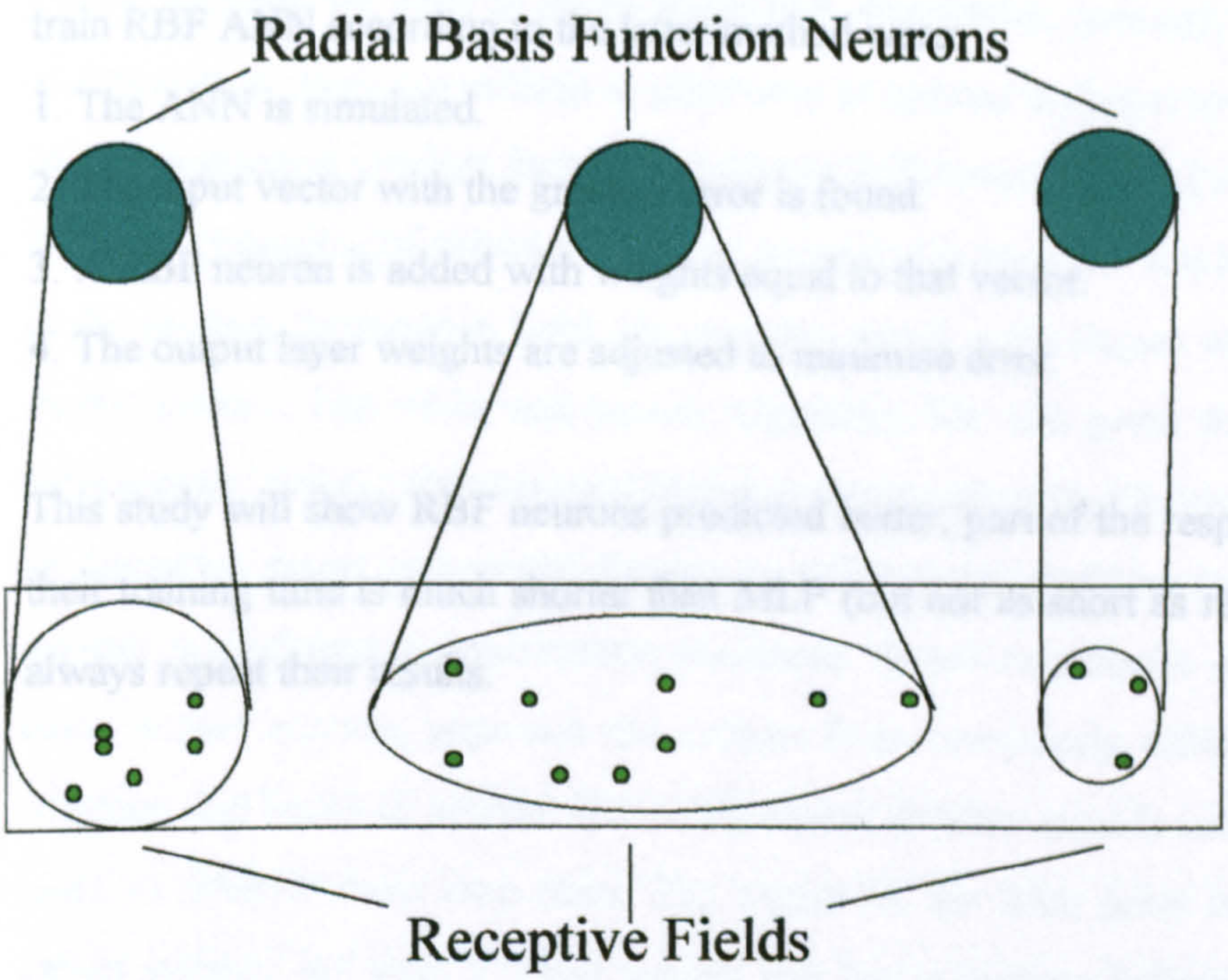


Figure 2.12: Receptive fields of each radial basis function (RBF) neuron.

The activation function of RBF neuron is a Gaussian one. This means that the closer the input is to the centre of the receptive field the more the output of the RBF neuron. The output of RBF neuron is calculated according to the following formula:

$g_j(x) = \exp[(-x-\mu_j)^2/o_j^2]$

Where x is the input vector, μ_j is the centre of the receptive field, o_j is the width of the receptive field and $g_j(x)$ is the output of neuron number j.

An important parameter to choose is the width of the receptive field. The receptive fields should overlap, but they should not be too big, to prevent them from being highly active for a single input. In this study the width of the receptive field is termed as spread constant. This study tried numerous spread constants and the change in them influenced immensely the generalisation ability of the ANN.

There is another important parameter and it is the number of RBF neurons. Two methods of choosing them were selected. One of them is that there are as many RBF neurons as there are input vectors (it is a vector since it could be composed of more than one independent variable). The second one is that neurons are added to the ANN until the sum

squared error falls below an error goal or a maximum number of neurons has been reached. The second method showed better results in this study. To summarise, the steps used to train RBF ANN according to the latter method were:

1. The ANN is simulated.
2. The input vector with the greatest error is found.
3. A RBF neuron is added with weights equal to that vector.
4. The output layer weights are adjusted to minimise error.

This study will show RBF neurons predicted better, part of the responses. Like regression, their training time is much shorter than MLP (but not as short as regression) and they will always repeat their results.

2.5 Expert systems

Expert systems are used in many fields like: electronics, geology, engineering, chemistry and medicine. They are utilised in pharmacy in general and also specifically in the domain of pharmaceutical product formulation (Rowe & Roberts, 1998). Rowe explains in his book the basic principles of expert systems in general. He surveyes a number of expert systems in the product formulation field. He explains which technologies were incorporated in the expert systems like ANN and genetic algorithm. He also gives the background to these technologies. Bohl (1990) brings detailed examples of solving formulation problems. When he discusses expert systems he focuses on subjects like building regression equations and solving multiobjective optimisation problems. These two books on product formulation using expert systems approach the subject from completely different angles. It is quite common that books or articles discussing expert systems used in solving the same problem seem so different from each other. The reason for the latter point stems from the fact that expert systems are such a broad subject that has numerous definitions. One of them was presented in the introduction. Main issues regarding expert systems (Giarratano & Riley, 1994) are given in the following paragraphs. The aim of this section is to give the reader a 'feel' for expert systems.

Characteristics of expert systems:

1.The quality of expert system advice should be better than or equal to a human expert in the field.

2. The response time of an expert system should not be too long.

3. Reliability - the expert system should not crash frequently otherwise people will not want to use it.

4. Explanation - it is important the expert system explain its decisions. It is hard to accept answers that can not be explained. It is also a way to track errors.

5. Flexibility - the expert system should have facilities to update the expert system knowledge.

The technology of ANN is considered as a milestone in the development of expert systems and these terms (ANN and expert systems) are quite often related. Bohl (1990) describes an expert system for formulation of cupcakes with the use of regression. Hence, ANN and also regression are technologies that are used in expert systems.

One important expert system is MYCIN (Gallant, 1993). This system diagnoses illness and suggests a remedy from the medical domain of bacterial infections. It was a milestone in the field of expert systems because for the first time the knowledge base was separated from the inference engine. Emptying the knowledge out of MYCIN created Expert-System shell called EMYCIN; the added E stands for essential or empty MYCIN.

There are computer languages like Prolog (Bratko, 1990) that are more suited for the creation of expert systems than the common languages, like Fortran, which is more suited to numeric calculations. The selection of computer language for the creation of the expert system is an important decision in expert system development.

Expert systems usually have some typical components. They have a user interface in order for the user to interact with the system, a help or advice facilities that will explain the expert system decisions, and a database of facts that could include properties of excipients as well as other relevant data on the tablet formulation domain. The other two componenets are an inference engine - the computer has to apply rules in response to database data, and a knowledge acquisition facility - the user must have the ability to enter knowledge to the system. For example, relevant to this study, it must have the ability to enter new excipients and their properties into the database.

Expert systems possess many advantages. They increase availability – an expert system can be put on many computers and used by many people simultaneously whereas a human expert is not always available. Once they are put on many computers the knowledge is already in them so costs are reduced since there is no need to pay consulting fees. The human expert may die and their knowledge vanishes whereas storage on computer makes the knowledge more permanent. Lai et al. (1996) programmed an expert system with the knowledge of several experts on capsule formulation. Several experts involved in developing an expert system can create a more powerful expert system than one expert can. Expert systems are more reliable than human experts are, they could sometimes generate a better solution than the human expert whose knowledge was put into the system. For example, in cases where there are stress conditions on the human experts that prevent them from thinking clearly about the problem (the input to the expert system was not given under conditions of stress). Providing the same explanation regarding a decision several times to

many people is a task that involves tedious repetition. On the other hand an expert system is never tired of giving explanations. Expert systems can be used as an intelligent tutor that explains expert knowledge. Expert systems can be used as a database from which the knowledge within can be retrieved easily; e.g. the user can ask the expert system about relevant excipient properties instead of retrieving the information by searching in many books/articles. Hence, it reduces literature search time. Fast response – e.g. in coating of tablets, if human expert is asked about adjusting coating process parameters according to problems arising in the process (troubleshooting), he/she can think about this without any critical time constraint and the expert knowledge will be entered into the expert system. If the expert encounters the problem on site a decision on the solution may be too late/slow. On the other hand the expert system could give a fast answer that would save the batch whilst in the coating phase.

3. Application of Artificial Neural Networks and Multivariate Regression to Solid Dosage Form Optimisation

3.1 Introduction

This chapter deals with the development and application of artificial neural networks (ANN) and multivariate regression analysis to pharmaceutical formulation. It is based on a study of tablet formulation and process variables. The data for this study was taken from Patel (1996).

Four problems were examined in this study. Is it possible to predict the best ANN methodology based on ANN trained on all data? This method of screening used by others (Hussain et al., 1994) could reduce the time required to perform validation experiments on each ANN topology. But still this method has to be checked for its validity. The second issue examined the significance of the validation method employed. As different studies have employed different validation methods it is important to examine whether the chosen validation method influences research results. As an example, Murtoniemi et al. (1994a & 1994b) chose a validation set of five samples (using only some of the samples) whereas Hussain et al. (1991) chose the leave-one-out validation method (using all the samples). The third issue tackled the subject of comparing the predictive ability of ANN versus regression. It is important to examine if the new ANN methodology (Hussain et al., 1991) gives an advantage over regression methodology (Schwartz et al., 1973), since the

methodology used influences not just the predictive ability of the model but also the performance of the optimisation step. The latter stems from the fact that the optimisation step relies on the model generated. This problem has been tackled many times before (Murtoniemi et al., 1994a & 1994b; Hussain et al., 1991). However, this study attempts to do it in a more rigorous way, no work has been reported using leave-one-out experiments on so many topologies and none trying so many regression models. The fourth question that was examined is if scaling of ANN is a real necessity and if so which scaling method is preferable. If scaling influences substantially the predictive ability it is worth testing it since improving the predictive ability means better optimisation hence, less experiments, less development time and more money saved (which is true for every technique that improves predictive ability).

The chapter begins by introducing the methodology followed by the results in 4 subsections. The first presents results of ANN trained on all data as a tool to predict the best topology. In this the error for each topology is used to determine the best topology chosen. This screening experiment, using all the data could eliminate the computation time required to perform validation experiments on each topology. The second and third result subsections give the results for ANN and regression predictive ability, using two different validation methods. These subsections address the subjects of comparison between ANN and regression performance, also as to whether there is influence by the validation method employed. The final part of the results explains the scaling methods employed for ANN.

3.2 Methods

The experimental data utilised was from laboratory experiments conducted by Patel (1996). Screening experiments were done. Two methods of partitioning the data for training and validation sets were employed. In ANN different training methods and topologies were tested. In regression different variable selection methods were employed. The data were scaled and the method of evaluating the prediction ability was by calculating the average percentage deviation (this is the same as the relative error in percent).

Tablets containing 500 mg of paracetamol were formulated using directly compressible paracetamol, croscarmellose sodium, type A (Ac-Di-Sol®) as disintegrant, magnesium stearate as lubricant, and made up to 600 mg of tablet with lactose as diluent. In generating the experimental data the compaction force, percent lubricant and percent disintegrant were manipulated. The levels were 6, 12 and 20 kN for the compaction force; 0.25, 0.5 and 1% for the lubricant; 1, 2 and 5% for the disintegrant. Table 3.1 presents these independent variables values. The appropriate responses were measured in order to build a model that should allow the prediction of dissolution rate, disintegration time, hardness, tensile strength, two friability tests, thickness and mean weight. These response values are presented in Table 3.2. The manipulation of the independent variables was based on a factorial design. Factorial design is an experimental strategy in which factors are varied together, instead of one at a time (Montgomery, 1997). However, the compaction force could not be absolutely fixed, but the actual value was monitored, and these values were used in the calculation of the regression and ANN models. The compaction force was nominally set at 3 different levels. The other independent variables were fixed at exactly 3 levels. In total 27 experiments were conducted by Patel (1996). For every formulation, 20 tablets each, were checked for tablet weight, thickness and hardness. Tensile strength data for the sample tablets were obtained from the tablet thickness and hardness. Two friability tests were conducted: impact friability and erosion friability. The friability tests were triplicated using 5 tablets in each trial. Disintegration time data were obtained using 6 tablets per formulation. Dissolution rate data were obtained using 3 tablets per formulation. In all data analysis the average values of these measurements were used. Tensile strength was the only response that was not measured directly, but calculated in a manner explained in the next section.

The tensile strength measured was diametrical tensile strength denoted by σ. It was calculated using the equation (Lieberman et al., 1990):

$$\sigma = \frac{2F}{Dt\pi}$$

whereas F is the maximum force to cause tensile failure (fracture). D is the diameter and t is the thickness. As an example, the radial tensile strength calculation of one sample out of 20 in case number 5 was:

$(2 \times 9.81\ \text{m sec}^{-2} \times 3.39\ \text{kg} \times 1000)/(12.1\ \text{mm} \times 5.65\ \text{mm} \times \pi) = 309.84\ \text{kN m}^{-2}$

whereas 3.39 is the tablet hardness, 9.81 is the accelaration constant (g) and is used for the trasformation to units of force. The multiplication by 1000 in order that the units will be in kN. 12.1 mm is the diameter and 5.65 mm is the tablet thickness.

Table 3.1: Independent variables manipulated

Case	Lubricant (%w/w)	Disintegrant (%w/w)	Force (kN)
1	0.25	5.00	6.48
2	0.25	2.00	6.44
3	0.25	1.00	6.22
4	0.50	5.00	5.94
5	0.50	2.00	6.00
6	1.00	5.00	5.73
7	1.00	1.00	5.96
8	0.25	5.00	11.82
9	0.25	2.00	13.98
10	0.25	1.00	12.12
11	0.50	5.00	11.27
12	0.50	1.00	11.48
13	1.00	1.00	12.12
14	0.25	5.00	20.45
15	0.25	2.00	19.38
16	0.25	1.00	20.21
17	0.50	5.00	19.94
18	0.50	2.00	19.68
19	0.50	1.00	20.18
20	1.00	5.00	19.35
21	1.00	2.00	19.65
22	1.00	1.00	19.71
23	0.50	1.00	6.26
24	1.00	2.00	6.13
25	0.50	2.00	12.42
26	1.00	2.00	12.05
27	1.00	5.00	11.17

Table 3.2: Responses measured.

Case	Mean Weight (mg)	Thickness (mm)	Hardness (kg)	Tensile Strength (kN/m^2)
1	598.9	5.43	3.71	353.31
2	600.9	5.49	4.05	380.87
3	610.0	5.55	3.97	369.31
4	599.1	5.56	2.74	253.77
5	619.0	5.67	3.44	313.21
6	594.3	5.54	2.52	234.99
7	632.1	5.75	3.11	278.91
8	588.7	4.88	7.52	795.40
9	607.2	4.91	9.20	966.96
10	609.1	5.02	7.42	763.08
11	606.2	5.06	6.25	637.41
12	618.8	5.15	7.04	705.41
13	594.7	5.17	6.99	697.36
14	593.7	4.62	12.39	1385.41
15	600.5	4.70	11.79	1294.09
16	619.5	4.83	12.54	1339.53
17	594.0	4.65	10.64	1180.66
18	600.3	4.72	10.56	1156.43
19	612.5	4.79	7.52	810.80
20	591.1	4.64	9.56	1062.81
21	626.3	4.91	10.32	1086.33
22	625.6	4.88	10.87	1150.93
23	617.9	5.63	3.38	310.11
24	622.5	5.67	2.97	270.07
25	606.2	4.99	7.45	771.80
26	628.0	5.17	6.92	690.09
27	621.5	5.19	6.18	614.34

Table 3.2 (cont.): Responses measured.

Case	Erosion Friability (%)	Impact Friability (%)	Disintegration Time (sec)	Dissolution Rate (k, mg/min)
1	1.66	2.21	115	36.22
2	1.46	1.96	102	55.27
3	1.71	2.03	65	73.38
4	2.02	2.59	128	25.72
5	1.66	2.19	102	58.45
6	2.10	2.79	116	30.37
7	1.86	2.50	58	133.47
8	0.77	1.04	24	46.89
9	0.68	0.91	22	72.30
10	0.81	1.18	33	65.10
11	0.90	1.12	39	98.15
12	1.03	1.06	34	62.65
13	1.01	1.05	49	89.75
14	0.50	0.67	58	72.48
15	0.51	0.73	50	75.67
16	0.51	0.69	93	56.79
17	0.69	0.65	59	62.35
18	0.65	0.78	61	54.15
19	0.61	0.71	93	39.89
20	0.72	0.81	59	72.82
21	0.62	0.91	78	59.48
22	0.66	0.85	98	46.59
23	1.71	2.36	58	77.18
24	1.94	2.52	71	107.11
25	0.78	1.14	31	77.45
26	0.93	1.14	42	97.35
27	0.92	1.20	46	94.76

Screening experiments using ANN were done using all the 27 data points to train various ANN topologies. The sum squared error (SSE) and the mean relative error (MRE) for each topology were recorded.

The data was partitioned into a training set (used for building the model) and a validation set (this process is called cross-validation) as follows:
Five data sets were selected randomly for validation and the remaining twenty-two data sets for building the model.
The second method was jacknifing. Jacknifing is the leave one observation out at a time approach (Mendenhall & Sincich, 1996). This method is also used to find influential observations in regression. There were 27 cases in our study, each time one observation was taken out and used for validation. The other 26 cases were used for building the model. This process was repeated 27 times, so in the end all the data was used for training and for validation.

ANN experiments were repeated 4 times using the first method of partitioning the data (4 times × 9 topologies = 36 networks to run) using different activation functions. The activation functions that were examined were:

1. Hyperbolic tangent sigmoid transfer function.
2. Log sigmoid transfer function.
3. Linear activation function.

Combinations used were (the first number is for the hidden layer neurons, the second is for the output layer neurons):

a. 1 and 1.
b. 2 and 2.
c. 2 and 3.
d. 1 and 3.

The last combination, which is hyperbolic tangent sigmoid transfer function in the hidden layer and linear activation function in the output layer, was regarded as the reference ANN. The activation function chosen was then employed in the second method of partitioning the data. All ANN were run for 100,000 epochs. Each epoch consisted of all the training set, and the weights were adjusted after each epoch. The latter method of updating the weights is called batch training, and is different from incremental training in which the weights are updated each time input is presented to the ANN.

The polynomial regression equations were derived using stepwise regression or backward elimination as methods of variable selection or without any variable selection at all. The process of finding the final equation is done by first choosing the order of the model and the interaction terms and after that employing the method of variable selection. Twelve models were derived for each method of partitioning the data. The different models are described in the Results & Discussion section. In the first method of partitioning the data (simpler than the second one, thus used here for demonstration purposes) from each model 8 different polynomial equations (8 response variables) were derived using the least squares method, e.g. a first order model with no interaction terms, $y = a_1*x_1 + a_2*x_2 + a_3*x_3$. Eight equations were derived using this model, one for each response variable. The values of the coefficients a_1, a_2 and a_3 thus differentiate between the equations.

The method of evaluating the prediction error was by calculating the MRE (MRE = $1/n * \Sigma ABS[(t-o)/t]$). The MRE is computed by taking the absolute value (ABS) of the difference between the desired target output that is the observed value (t) and the attained output (o) as a fraction of the desired target output (t), and summing across all trials (Σ). The sum is divided by the number of trials (n) to get a mean value (Masters, 1993). In this thesis all MRE values are presented in percentage (multiplied by 100) unless stated otherwise.

The different scaling used is best explained using the simple ideas of matrix manipulation. Define **P** as a matrix of all data input. Each column **y** in that matrix thus represents one input variable, whereas each row **x** represents the 3 input scalars for a specific data set. In the same way we define **T** as a matrix of outputs. Each column **y** in that matrix represents one output variable, whereas each row **x** represents the 8 output scalars for a specific data set. The scalar value of each one of the 27 members of a column is represented by y.

Scaling the data for the ANN was done using 4 different methods:

1. Set the highest value in **y** to be 0.9. This is done by finding the maximum value in each **y**. And the operation $y_{new} = y_{old} / \max(\mathbf{y}) * 0.9$ for the appropriate max(**y**) scalar value according to the relevant response. Where max(**y**) is the maximum value of each output variable (i.e. one maximum value for each response).

2. Set the values between 0.1 and 0.9. The values max(y) and min(y) are the maximum and minimum of y respectively. Define $A_{max} = 0.9$ and $A_{min} = 0.1$ as the ANN practical limits. Define $r = (A_{max} - A_{min}) / (max(y)-min(y))$. The translation of each *y* value in the column is done using the following formula: $y_{new} = r * (y_{old} - min(y)) + A_{min}$ (Masters 1993; Smith, 1993).

3. We can scale each *y* by subtracting its mean (denoted by mean(y)) and dividing by its standard deviation (denoted by std(y)). The formula is: $y_{new} = (y_{old} - mean(y)) / std(y)$. The mean and standard deviation were estimated from each vector y. This method (zscore) ensures that 2 researchers using different units for the same measurement will get the same results (Masters, 1993; Bishop, 1996).

4. Methods 2 and 3 can be incorporated to one formula:

$y_{new} = r * ((y_{old} - mean(y)) / std(y) - min(y)) + A_{min}$

$= r / std(y) * y_{old} + (A_{min} - r * (mean(y) / std(y) + min(y))$ (Masters, 1993).

There were also ANN run without any scaling on the data (Method 5). ANN were trained using the leave one out method for validation and using the Levenberg-Marquardt algorithm in the backpropagation. The neurons of the hidden layer had a hyperbolic tangent sigmoid transfer function and those of the output layer had a linear activation function. A hundred epochs were run on each topology. The 9 different topologies examined were composed of 1 to 9 neurons in the hidden layer. The total number of ANN trained was thus 234 (i.e. 27×9) for each scaling method. To generate these predictions 5 steps were involved: scaling of input and output data, run ANN, simulate ANN, unscale ANN predictions and errors calculations.

For ANN, the software for programming ANN algorithms employed was MATLAB® with Neural Network Toolbox. SPSS® version 7.5 was used for the regression analysis procedures. The hardware employed was an IBM compatible personal computer (Pentium II, 150 MHZ/32 MB RAM).

3.3 Results & discussion

3.3.1 Screening experiments

For all the topologies that were considered the sum squared error (SSE) and the mean relative error (MRE) are tabulated in Table 3.3. The column on the left ("Model") shows the ANN topology, from one to nine hidden neurons. The two columns on the right show the SSE and MRE error values. The data of Table 3.3 is plotted in Figure 3.1. On the y-axis is the error term that is MRE or SSE and on the x-axis is the number of hidden neurons.

Table 3.3: The dependency of the SSE and MRE on the number of hidden neurons.

Model	SSE	MRE
318	2.3231	15.8815
328	0.9672	10.1564
338	0.6455	8.3344
348	0.5321	6.9497
358	0.3648	6.1582
368	0.3306	5.5620
378	0.3299	6.4498
388	0.2263	4.8715
398	0.2209	5.1670

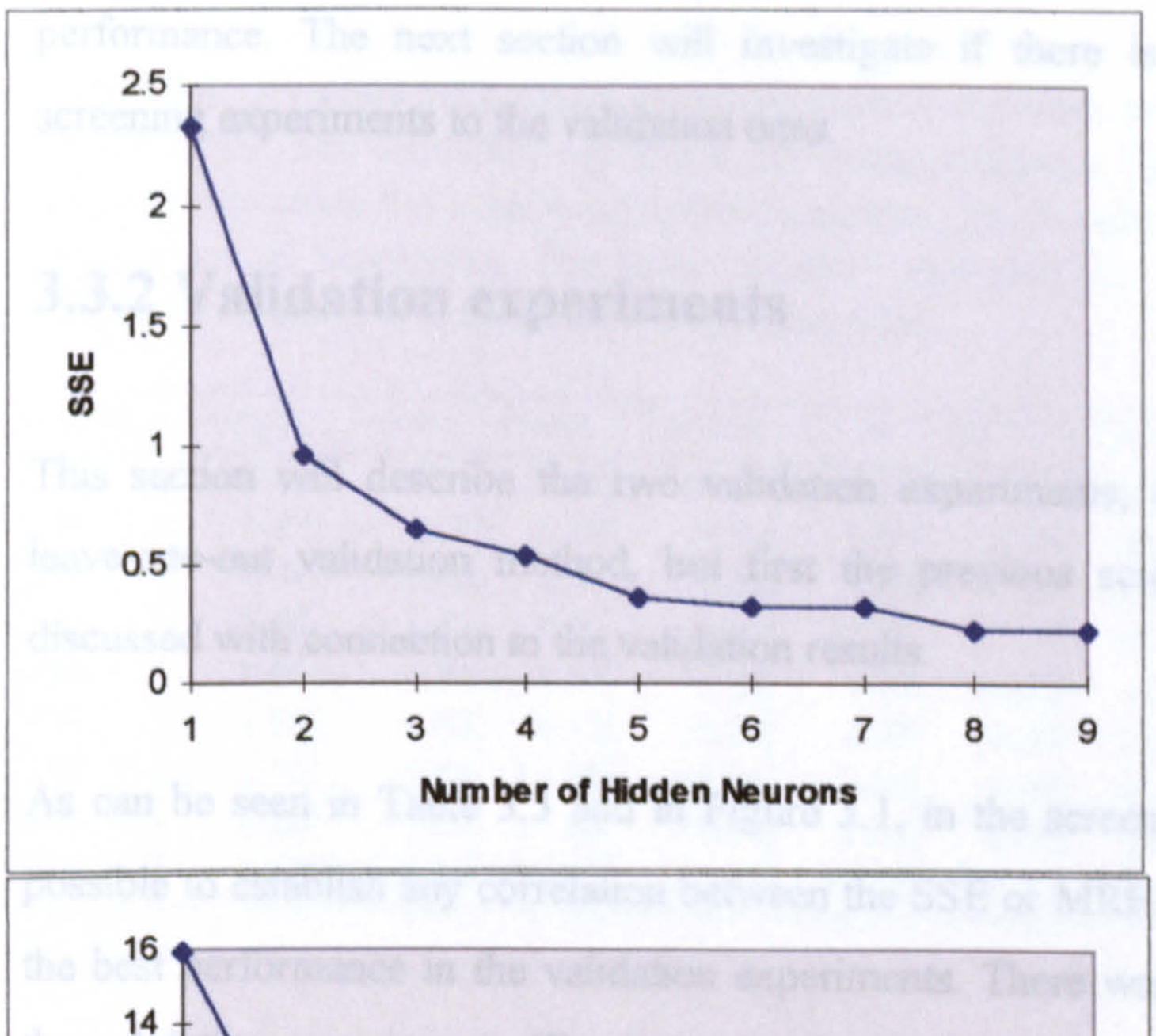


Figure 3.1: The two graphs above show the dependency of the SSE and MRE on the number of hidden neurons.

The reasoning of taking two error measurments is that in previous studies, the error was measured by two different methods. Previously the root mean square error and the SSE have been employed (Hussain et al., 1994; Jogarao et al., 1995), which are essentialy the same, whilst I have used other more distinct measures of errors. Both research teams conducted screening experiments for selecting the best topology before the validation step. Both used the time consuming (leave-one-out) validation test and saved time by the approach of not performing validation tests on all topologies. This study tried to identify if there was a correlation between the results of this approach to validation on all topologies. From the two graphs in Figure 3.1, the slope of the SSE/MRE reached a plateau after 6 hidden neurons suggesting that the topology of 6 hidden neurons will show the best overall

performance. The next section will investigate if there is a correlation between the screening experiments to the validation ones.

3.3.2 Validation experiments

This section will describe the two validation experiments, the cross-validation and the leave-one-out validation method, but first the previous screening experiments will be discussed with connection to the validation results.

As can be seen in Table 3.3 and in Figure 3.1, in the screening experiments, it was not possible to establish any correlation between the SSE or MRE to the topology that showed the best performance in the validation experiments. There was no correlation with any of the validation experiments. The reasoning for the latter statement is that the overall best ANN in the cross validation experiments was with 5 hidden neurons (see Table 3.4), and in the leave-one-out validation method the best ANN was with 3 hidden neurons (see Table 3.7) whereas the screening experiments found the best ANN was with 6 neurons in the hidden layer. The results were quite surprising since the method of screening is commonly practiced by researchers in the field (as was mentioned before: Hussain et al., 1994 or Jogarao et al., 1995).

The first approach used cross validation, splitting the data into two sets, 22 for training and 5 for validation. The MRE values of the validation sets for ANN and regression are shown in Tables 3.4 and 3.5 respectively. Looking at Table 3.4 and Table 3.5 the left-hand column is headed "Model". In Table 3.4 this heading stands for the topology of the ANN. For example, "348" means an ANN with 3 input, 4 hidden and 8 output neurons. In Table 3.5 the heading "Model" stands for the type of regression model. Columns two to nine give the MRE for each of the responses monitored. The average MRE for all the responses is given in the column on the right. The average MRE ranges from about 6% to a maximum of about 25% in the regression and ANN tables. MRE of the best models for both the regression and ANN are given in Table 3.6.

Table 3.4: Average percentage deviation (MRE) between predicted and observed values for the ANN. The validation method is the cross-validation. The middle number in the first column represents the number of hidden neurons. The number in bold represents the best result from all topologies. The term "Model" indicates the topology, e.g. "348" means ANN with 3 input, 4 hidden and 8 output neurons.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
318	2.05	1.98	14.47	19.17	7.58	7.77	59.92	30.29	17.90
328	1.67	1.05	6.34	9.19	8.97	8.43	26.08	**7.99**	8.72
338	1.84	**0.68**	**2.41**	**3.30**	11.63	11.24	25.38	20.37	9.61
348	1.09	1.61	8.01	8.06	15.74	17.79	15.73	9.89	9.74
358	1.51	1.01	4.42	4.25	7.69	10.95	**8.39**	16.97	**6.90**
368	1.56	1.65	9.12	10.85	**5.47**	10.25	36.97	16.38	11.53
378	2.12	1.74	11.68	14.21	6.24	7.63	17.21	20.67	10.19
388	**0.60**	1.18	11.45	12.93	7.82	**6.42**	18.33	31.45	11.27
398	2.68	1.56	25.38	27.87	11.88	12.43	79.07	40.71	25.20

Table 3.5: Average percentage deviation (MRE) between predicted and observed values for the regression analysis. The validation method is the cross-validation. The number in bold represents the best result from all regression models.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
1	2.04	1.36	5.72	6.72	7.79	12.41	18.54	36.71	11.41
2	2.94	1.80	**3.26**	**6.19**	10.55	10.98	25.06	13.09	9.23
3	2.37	1.79	6.00	7.74	8.86	10.64	22.69	5.76	8.23
4	2.04	1.10	7.36	6.63	7.79	11.34	23.19	**4.79**	8.03
5	1.94	1.16	11.45	15.15	9.00	12.35	18.45	8.70	9.78
6	1.82	1.06	9.31	8.08	7.79	10.82	17.75	20.05	9.58
7	**0.60**	**0.24**	14.12	17.28	**4.17**	**3.33**	**5.80**	7.70	**6.66**
8	2.04	1.06	9.31	8.08	7.79	10.82	17.75	15.99	9.10
9	0.60	0.30	15.08	18.41	4.20	4.54	6.37	6.93	7.05
10	1.71	1.60	12.55	11.37	33.24	31.77	60.70	17.94	21.36
11	2.04	1.60	8.50	8.42	31.17	29.34	60.62	20.05	20.22
12	1.56	1.60	8.69	8.56	30.63	28.80	70.90	23.18	21.74

Details of the models

1. Reduced six order model by stepwise regression (excluded terms greater than x^2). Criterion for enter was $p<0.05$ (refer to section 1.3 regarding p-value) and for removal $p>0.1$. If none of the independent variables succeed to enter the equation the criterion was changed for $p<0.1$ for enter and $p>0.11$ for removal (p for enter must be smaller than p for removal). And if the problem persisted the values checked were $p<0.15$ for enter and $p>0.16$ for removal and so on in steps of 0.05.
2. Third order model not included terms in the power of 3 (excluded x^3 terms).
3. Reduced third order model (excluded x^3 terms) by backward elimination $p>0.1$.
4. Reduced third order model by stepwise regression (excluded x^3 terms). Stepwise regression was done as in 1.
5. Second order model + interaction term- $x_1*x_2*x_3$.
6. Second order model + interaction term- $x_1*x_2*x_3$ reduced by backward elimination $p>0.1$.
7. Second order model.
8. Reduced second order model by backward elimination $p>0.1$.
9. Reduced second order model by stepwise regression. Stepwise regression was done as in 1.
10. First order model + interactions terms (x_1*x_2, x_1*x_3, x_2*x_3, $x_1*x_2*x_3$).
11. First order model + interactions terms (x_1*x_2, x_1*x_3, x_2*x_3, $x_1*x_2*x_3$) reduced by backward elimination.
12. First order model.

Table 3.6: Average percentage deviation between predicted and observed values for the best models of regression analysis and ANN. The validation method is the cross-validation.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
Reg.	0.60	0.24	3.26	6.19	4.17	3.33	5.80	4.79	3.55
ANN	0.60	0.68	2.41	3.30	5.47	6.42	8.39	7.99	4.41

The MRE of the second validation method of the leave-one-out, are given in Table 3.7 and Table 3.8 for ANN and regression respectively. For both Tables 3.7 and 3.8, the left-hand column stands for the ANN topology and the regression model respectively. Columns two to nine give the MRE for each of the responses monitored. The average MRE for all the responses is given in the right-hand column. The best models for both the regression and ANN are given in Table 3.9.

Table 3.7: Average percentage deviation (MRE) between predicted and observed values for the ANN. The validation method is the leave-one-out. The middle number in the first column represents the number of hidden neurons. The number in bold represents the best result from all topologies.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
318	1.94	1.93	12.50	15.14	10.41	8.89	49.23	38.62	17.33
328	1.90	1.83	12.53	15.80	10.12	**8.32**	30.30	29.78	13.82
338	1.74	1.48	**9.37**	**11.54**	9.08	9.08	20.86	**25.60**	**11.09**
348	1.63	**1.19**	9.89	11.91	9.83	9.60	17.70	28.37	11.26
358	1.71	1.38	11.17	13.67	10.87	11.18	**15.07**	28.91	11.74
368	**1.55**	1.28	13.49	15.20	9.49	8.93	19.42	26.41	11.97
378	1.76	1.49	15.78	18.70	9.20	10.88	17.22	31.22	13.28
388	1.97	1.85	16.91	19.86	**7.85**	9.10	15.97	32.48	13.25
398	1.74	1.32	18.19	19.84	10.21	11.00	19.11	25.90	13.41

Table 3.8: Average percentage deviation (MRE) between predicted and observed values for the regression analysis. The validation method is the leave-one-out. Details of the models are the same as in Table 3.5. The number in bold represents the best result from all regression models.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
1	**1.28**	**0.88**	7.81	7.53	7.63	8.35	**9.13**	35.57	9.77
2	2.97	1.63	15.15	15.99	12.03	7.42	13.73	31.36	12.54
3	1.61	0.89	**7.65**	**6.99**	**7.25**	**5.64**	9.36	**19.28**	**7.33**
4	**1.28**	**0.88**	9.82	8.92	7.63	8.35	9.36	33.05	9.91
5	2.15	1.37	10.07	10.73	12.11	11.75	24.17	26.66	12.38
6	1.29	**0.88**	8.53	9.14	7.63	8.99	19.90	23.45	9.97
7	1.78	1.12	10.87	11.87	10.90	10.26	22.75	26.29	11.98
8	1.29	**0.88**	8.53	9.14	7.63	8.99	19.90	22.05	9.80
9	**1.28**	**0.88**	**7.65**	9.05	7.63	8.35	26.58	35.85	12.16
10	1.84	2.15	11.70	10.54	30.56	33.90	65.21	30.20	23.26
11	**1.28**	1.80	11.24	10.56	22.16	23.71	54.24	26.48	18.93
12	1.08	1.54	8.49	8.70	20.12	21.57	48.39	31.87	17.72

Table 3.9: Average percentage deviation (MRE) between predicted and observed values for the best models of regression analysis and ANN. The validation method is the leave-one-out.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
Reg.	1.28	0.88	7.65	6.99	7.25	5.64	9.13	19.28	7.26
ANN	1.55	1.19	9.37	11.54	7.85	8.32	15.07	25.60	10.06

As can be seen in the two validation experiments, different methods of validation gave different results. The leave-one-out method detected MRE that are greater in all the response variables (in ANN as well as in regression) than in the first experiment, using 22 samples for training and 5 samples for validation. The high MRE in the leave-one-out validation method is partly due to outliers, partly to chance and partly to extrapolation that is unavoidable in the jacknife experiment. The data for validation in the first experiment was in the domain of the training data so there was no need for extrapolation.

Generally, in the two validation experiments, predictions based on regression analysis method were better. The superiority of regression analysis was more obvious in the second experiment than in the first one. The second experiment is probably more important than the first one because it takes into account all the data for testing and for validation.

The test was not equivalent because the ANN took into account all the relationships between the input variables and the output variables and did not simply focus on a single response variable as in regression. To conduct a fair test one should also model ANN with one output only, at the expense of losing ANN inversion capability as a multiobjective optimisation method (Zell et al., 1994)

Whether using ANN or regression one can choose the topology/model (in ANN one optimum model for all response variables, and in regression one optimum method for deriving the regression equations) for prediction with an overall error higher than the best model but better for predicting the important parameters such as dissolution rate at the expense of overall predictive ability. For example, in the first experiment we can choose the topology with two hidden neurons and hence increase the overall MRE from 6.90 in the 5 hidden neurons (the best model) to 8.72, but decrease the MRE for the important parameter of dissolution from MRE of 16.97 in the 5 hidden neurons to 7.99 in the two hidden neurons.

Selecting the best overall model is important in ANN because it is possible to do multiobjective optimisation using one ANN. In regression the same method of selecting coefficients can be used for all responses but there is no advantage in using the same method since it would yield 8 different equations, one for each response. Hence, in regression there is no one model that captures all independent variables and responses that could be used in multiobjective optimisation. In either case it was not possible to say which ANN/regression model is the best from the two experiments because the answer changed when the method of partitioning of data was changed. Since it seems that the second method of partitioning the data (jacknife) was superior, putting all the data in an ANN of 5 hidden neurons (best topology of the leave-one-out validation method), training the net and performing multiobjective optimisation seems to be the best method of optimisation. Multiobjective optimisation will be discussed later.

The MRE of regression leave-one-out validation method presented in Table 3.8 show variable selection on third order polynomials gave the best models except for the response of disintegration time. Examination of the regression coefficients for modelling of the disintegration time response (details of the regresssion equations are given in Appendix A) show that the selected coefficients did not include terms greater than third order but included one x^3 coefficient. The third order polynomials did not included these terms. These terms were included in a higher polynomial described in Table 3.5 (model 1). To see if there was a need in higher polynomial than a third order one, additional polynomial of third order that included x^3 terms was build for modelling the disintegration time response and stepwise regression was done on this polynomial. It generated the same regression model as the one that was built with higher polynomial model (model 1). Hence, it seems there is no need in modelling with polynomials greater than third order. Since all the best models are variations of third order polynomials there is a need to use polynomials greater than second order, which is the common approach (see Introduction).

3.3.3 Scaling

The scaling experiments are shown in Table 3.10. The column on the left represents the scaling method. Each scaling method is explained in detail in Section 3.2. Columns 2 to 9 present the MRE and the column on the right represent the average MRE of all 8

responses.

Table 3.10: MRE results when ANN were trained on scaled data according to methods 1-4, whereas method 5 is without scaling.

Model	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.	Average
1	1.42	1.26	8.77	10.82	8.61	8.30	15.53	22.10	9.60
2	1.47	1.17	9.49	11.63	8.95	8.42	16.18	24.01	10.17
3	1.56	1.35	9.34	11.05	9.31	8.84	16.61	22.42	10.06
4	1.47	1.17	9.49	11.63	8.95	8.42	16.18	24.01	10.17
5	1.75	4.74	27.51	30.54	32.94	36.29	45.67	24.14	25.45

If ANN has an activation function that is linear in the output layer, scaling is not necessary (Masters, 1993), but as can be seen in the Results & discussion section, scaling improves prediction ability even in this type of ANN. In feedforward ANN there are no limits to the input values. Nevertheless, when there is in use training algorithms that minimise the total error of all outputs, like in this study, if the output variables are unequally scaled, those with larger variability will be favoured, as they will dominate the error sum (Masters, 1993). For example, it is obvious that the output variable dissolution rate (large variability) will dominate the error sum relative to the mean weight variable (small variability). If one scales the input variables to the same range the ANN does not have to learn the magnitude of the input weights by assigning large weights to small variables and small weights to large variables. For cases where the algorithm imposes limits on the size of the weights the system could get stuck as a consequence of not scaling the input data. In the data of this study, the difference between the values of the input variables is of just one order, so this problem was not significant. Nevertheless, it seems reasonable to try scaling the data of both input and output variables to improve results.

Looking at Table 3.10, scaling with method 5, the raw data without any change, it is obvious that scaling is necessary from looking at the big average MRE related to this method. Method 1, that is the simplest one for scaling, seems to be the appropriate one for this set of data since it has the smallest average MRE. Hence, this method was chosen in this study. Examination of the variability of the data scaled according to method 1, of the two output variables, mean weight and dissolution rate, will yield range between extremes of 0.838 to 0.9 and 0.173 to 0.9 respectively. The gap between the extremes in the first case is 0.062 and in the second case is 0.73. So there is one order difference between the range (which is a measure of variation) of the two different output variables. But the neurons of the output layer adjusted the weights and learned this variability without any

problem. To summarise, as opposed to other studies that did not examine the scaling aspect, this study explored this technical issue and found it could be more important than optimising the different variables related to ANN (like topology).

3.3.4 Getting and visualising the regression equation

Looking in the first three cases in Table 3.1 & Table 3.2 one could see that in the independent variables only disintegrant level changed (the compaction force is more or less fixed). The disintegration time and dissolution rate of these cases shows that, as the disintegrant level decreases there is a decrease in disintegration time and increase in dissolution rate. This is for the lowest compaction force level and lowest lubricant level. Logically, quite the opposite is the expected trend (regarding drug dissolution). However, looking just at a few cases and trying to analyse them could be misleading so it is better generate a plot that will throw light on all the cases. Such a plot was generated in Figure 3.2. From Figure 3.2 one could see that when compaction force is fixed to medium level increasing disintegrant concentration level by a factor of five from 1 to 5% causes just slight increase in dissolution rate. As opposed to this, for lubricant levels the trend is reversed at a certain level and there is not one continuous trend. Till a certain level adding lubricant causes the dissolution rate to deteriorate (a decrease in dissolution rate) whereas at a higher lubricant level, addition of lubricant reversed the trend. This is probably due to the fact that for lower levels of lubricant its dominant property that influences dissolution rate is its hydrophobicity and at higher levels it causes the tablet to be weaker hence in the presence of water the tablet dissolve faster. So it has micro and macro effects that act in an opposing manner on the dissolution rate response. Since visualising the data is of much importance as was discussed earlier, generating the regression equation (imperative in order to try predicting the experimental result, optimise or visualise the data) and visualising the data according to the equation will be discussed in the next section.

Tables 3.11 and 3.12 describe how in a stepwise manner the best regression equation for the dissolution, in the first experiment (cross-validation), was derived. In Table 3.12 it can be seen that the final equation gave a R^2 value of 0.716, this means that 71.6% of the sample variation in Y can be explained by using X to predict Y (Mendenhall & Sincich, 1996). Table 3.11 demonstrates each step in the stepwise regression variable selection method to arrive at the final equation ($Y = 0.512 - 1.315*x_2 + 3.805*x_2*x_3 + 0.455*x_1^2 -$

$0.860*x_3^2*x_1 - 2.436*x_3^2*x_2$). Notice the significance level of the coefficients, as this is the factor used for their selection. Value of 0.019 for significance level means there is probability of 1.9% that the term ($x_3*x_3*x_2$ in the last step) is zero (that the null hypothesis is correct). The global significance level is 0.001 means that there is probability of 0.1% that all the model coefficients are zero, or in other words, there is probability of 99.9% that at least one of the model coefficients is non-zero.

Table 3.11: The process of variable selection, which used the stepwise regression methodology, for the derivation of the best dissolution equation in the first experiment.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.491	.062		7.880	.000
	X2	-.133	.107	-.268	-1.245	.227
2	(Constant)	.541	.064		8.448	.000
	X2	-.415	.178	-.839	-2.334	.031
	X2X2X3	.459	.239	.690	1.921	.070
3	(Constant)	.739	.095		7.745	.000
	X2	-.743	.202	-1.503	-3.685	.002
	X2X2X3	.959	.286	1.442	3.352	.004
	X3X3	-.323	.125	-.624	-2.573	.019
4	(Constant)	.698	.097		7.164	.000
	X2	-.806	.202	-1.631	-3.997	.001
	X2X2X3	-.313	.955	-.470	-.327	.747
	X3X3	-.603	.235	-1.166	-2.561	.020
	X2X3	1.534	1.101	2.257	1.392	.182
5	(Constant)	.714	.083		8.609	.000
	X2	-.809	.197	-1.637	-4.117	.001
	X3X3	-.548	.161	-1.060	-3.410	.003
	X2X3	1.189	.314	1.750	3.790	.001
6	(Constant)	.681	.084		8.109	.000
	X2	-.813	.191	-1.644	-4.245	.001
	X3X3	-.563	.157	-1.088	-3.587	.002
	X2X3	1.206	.306	1.776	3.947	.001
	X1X1	.116	.082	.232	1.406	.178
7	(Constant)	.616	.083		7.434	.000
	X2	-.814	.175	-1.647	-4.659	.000
	X3X3	-.275	.198	-.533	-1.390	.183
	X2X3	1.214	.279	1.787	4.351	.000
	X1X1	.329	.127	.661	2.602	.019
	X3X3X1	-.587	.280	-.793	-2.097	.052
8	(Constant)	.565	.077		7.306	.000
	X2	-1.332	.278	-2.694	-4.783	.000
	X3X3	-.160	.184	-.310	-.870	.398
	X2X3	3.665	1.120	5.394	3.273	.005
	X1X1	.398	.117	.800	3.399	.004
	X3X3X1	-.703	.255	-.950	-2.752	.015
	X3X3X2	-2.199	.980	-3.007	-2.245	.040
9	(Constant)	.512	.047		10.789	.000
	X2	-1.315	.276	-2.660	-4.771	.000
	X2X3	3.805	1.100	5.600	3.460	.003
	X1X1	.455	.096	.915	4.737	.000
	X3X3X1	-.860	.179	-1.162	-4.794	.000
	X3X3X2	-2.436	.934	-3 332	-2.609	.019

a. Dependent Variable: DISSOL

Table 3.12: Summary of the models, which were examined in the stepwise regression methodology, for the derivation of the best dissolution equation in the first experiment.

Model Summary[j]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.268[a]	.072	.026	.15850327
2	.472[b]	.223	.141	.14881554
3	.657[c]	.432	.337	.13073345
4	.700[d]	.490	.370	.12745125
5	.698[e]	.487	.401	.12425034
6	.735[f]	.540	.432	.12101302
7	.800[g]	.639	.527	.11047635
8	.854[h]	.730	.622	9.87E-02
9	.846[i]	.716	.628	9.80E-02

a. Predictors: (Constant), X2

b. Predictors: (Constant), X2, X2X2X3

c. Predictors: (Constant), X2, X2X2X3, X3X3

d. Predictors: (Constant), X2, X2X2X3, X3X3, X2X3

e. Predictors: (Constant), X2, X3X3, X2X3

f. Predictors: (Constant), X2, X3X3, X2X3, X1X1

g. Predictors: (Constant), X2, X3X3, X2X3, X1X1, X3X3X1

h. Predictors: (Constant), X2, X3X3, X2X3, X1X1, X3X3X1, X3X3X2

i. Predictors: (Constant), X2, X2X3, X1X1, X3X3X1, X3X3X2

j. Dependent Variable: DISSOL

As was demonstrated earlier, it can be useful to plot the data in order to understand the relationship in the system between the independent variables and the dependent variables. Figure 3.2 shows a plot based on the regression equation, the derivation of which was explained earlier in this section. Since there are 3 input variables and 8 response variables, it is very informative to see 8 plots, one for each response variable. The problem is that this is not possible since this requires plotting a 4-dimensional plot and that is not feasible. It is possible to describe the data in a 3-dimensional plot by fixing one input variable at a specific value. Under experimental conditions it was not possible to fix the compaction force exactly to only 3 values but it is possible to derive regression equation and to fix the compaction force to any desired level in order to plot a 3-dimensional picture.

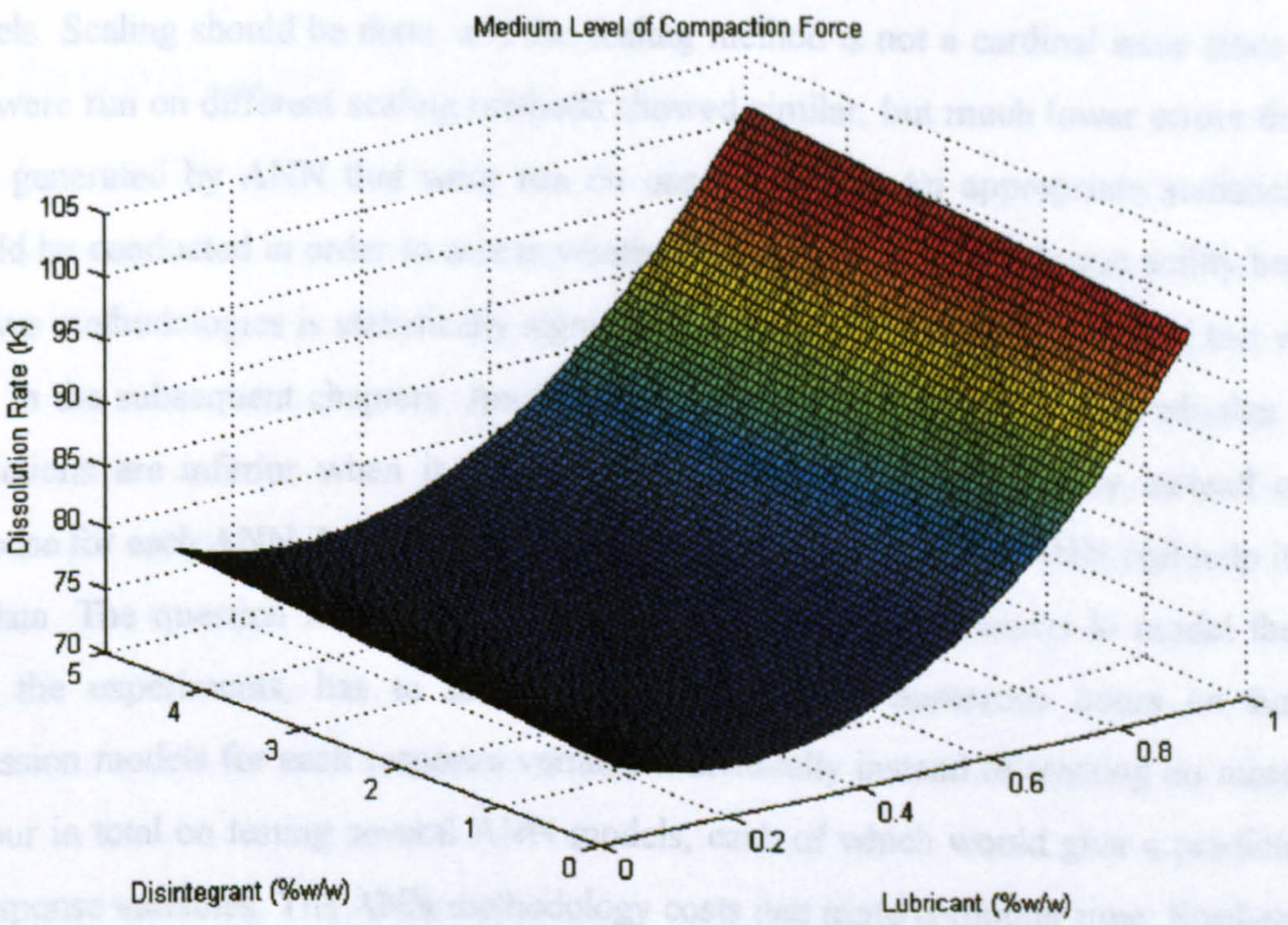


Figure 3.2: Plot of dissolution rate (mg/min) against disintegrant and lubricant levels (%w/w) whereas the compaction force is fixed to medium level of 12 kN. The regression equation being represented in this plot is derived from the training set of the first experiment.

3.5 Conclusions

Generally, the ANN has been found to be slightly less effective than regression approach since regresssion predictions showed lower MREs for all 8 responses in the leave-one-out validation method. The validation results depend on the validation method and it is worth considering the method of validation since the model chosen is used for optimisation (if the model is not good the optimisation performed on this model will be useless). It is not possible to predict the validation results by screening experiments since there is no correlation between the best ANN topology found in these experiments to the ones found by the validation experiments. It is worth using polynomials greater than second order, which is the common approach (see examples of modelling with second order polynomials in section 1.7), since variations on third order polynomials generated the best regression models. Scaling should be done, and the scaling method is not a cardinal issue since ANN that were run on different scaling methods showed similar, but much lower errors than the ones generated by ANN that were run on unscaled data. An appropriate statistical test should be conducted in order to assess whether the difference in prediction ability between the two methodologies is statistically significant or insignificant. This statistical test will be done in the subsequent chapters. Another issue that has to be addressed is whether ANN predictions are inferior when it predicts eight responses simultaneously instead of one response for each ANN. It is possible that the reduced complexity of ANN will help it learn the data. The question that the pharmaceutical researcher, who wants to model the data from the experiments, has to ask is whether to spend numerous hours on building regression models for each response variable individually instead of wasting no more than an hour in total on testing several ANN models, each of which would give a prediction of all response variables. The ANN methodology costs just more computer time. Furthermore, it may be that ANN might prove to be better for problems with a more complex hypersurface.

4. A Rigorous Statistical Comparison of ANN & Regression Models

4.1 Introduction

There are several questions that the current study attempts to answer. Does an ANN simulate better if its complexity is reduced? The complexity is to be reduced by not modelling all the system at once but modelling with several ANN each one having only one output neuron. No worker has addressed this important question before. Another question that remains unanswered and that has not been addressed before was whether the difference in prediction between ANN and regression is statistically significant. No study has presented the prediction errors of regression versus those of ANN with a check to see if the difference between these prediction errors is significant. A rigorous statistical comparison between ANN and regression errors was conducted. The underlying data used for the statistical comparison between ANN and regression was the same as that used in Chapter 3. Knowing one method is better than the other, even if the difference is statistically significant does not mean it is preferable to use the more accurate method. The reason for this is that the accuracy of prediction might depend upon the response value. For example, it may be that ANN have lower average error than regression but it is preferable to use regression in the domain of response values where it predicts better than ANN. The same reasoning applies when asking the question, which method predicts better at the extreme values of the responses? This was also tackled in this chapter.

Why then is it important to test whether ANN is more precise than regression? In an analogy to the game of darts, the darts being thrown are the predictions. The closer the darts are to the target the more accurate are the predictions. The scatter of the darts represents the precision term. It is important since it might be that a method is more accurate but it is preferable to use the less accurate method that is more precise. For example, if the mean relative error value (MRE) in one method is 12% compared to 10% with the more accurate method but in the latter method the predictions are much more scattered it might be better to choose the less accurate method. The less accurate method might be preferable since there is less risk of getting a large prediction error.

The question of whether the model is biased was also examined. For example, if regression equation yields predictions that are almost always greater than the observed value it is possible to trace this in the bias test. After tracing the problem the bias may be reduced by lowering the intercept term of the equation (i.e. $y = a*x + b$, b is the intercept term) to yield lower prediction values that are closer to the observed values.

This study also focuses on a number of training methods to determine whether they offer any improvement on the commonly used methods.

This chapter opens by explaining the various statistical methods for the comparison between ANN and regression. This is followed by a detailed section of Results & discussion that will include a summary of the results from the best ANN models compared to the results of the regression models developed in the previous chapter. This includes all the comparison data between ANN and regression prediction ability. There will also be a description of optimisation of the error goal parameter (mean squared error criteria to stop training in ANN) for the dissolution rate response.

4.2 Methods

ANN were trained using various algorithms, including different types of backpropagation methods as well as training methods that are not of the backpropagation type. The neurons of the hidden layer were of the hyperbolic tangent sigmoid transfer function and those of the output layer had a linear activation function. Topologies were varied from 1 to 9 neurons in the hidden layer. The validation method employed was the leave-one-out method. The number of epochs was also varied. Total number of ANN run was 27×9 = 243 (9 for each topology and 27 for each validation cycle) for each unique combination of training method and number of epochs. In the radial basis function ANN, the variables being manipulated were the spreading constant and error goal.

The prediction sets of ANN versus regression were compared using various statistical tests. Each set was composed of 27 rows representing prediction for that case when excluded from the model (leave-one-out) and number of columns as the number of responses

Both ANN & regression input and output data were scaled according to method 1 described in section 3.2. In this method the highest possible value is 0.9.

A one tailed, paired t-test was conducted to test the hypothesis that the MRE of ANN is smaller than that of regression. Although the t-test does not require that the sample comes from a perfectly normal distribution (Norusis, 1997) it is still necessary that the population resembles a normal distribution. Hence, examination of the data was done before the t-test. For examination of the normality assumption the relevant histograms were plotted overlayed with a perfect normal distribution line. There were also formal statistical tests that were conducted in order to help decide if the data for the t-test came from a normally distributed population. The higher the significance value of Shapiro-Wilk statistical test is, the higher the possibility that the distribution of the data is normal, since the null hypothesis of this test is that the population distribution is normal. This test has been tabulated for sample sizes ≤ 50 (Shapiro & Wilk, 1965). The paired t-test was used to test if the difference between the MRE of ANN and regression was different from zero. It generated for each pair of ANN and regression columns, a new column of the differences between the MRE, and it tested if the mean difference was not equal to zero. Hence, the tests of

normality were done on these new columns of difference.

Bivariate correlations were carried out using the data sets of the predicted values of ANN and regression and the observed response values. All 3 possible correlations between them were examined for each response variable separately. The correlation coefficients that were examined are based on the Pearson and Spearman's correlations. The first uses the actual data and the second uses non-parametric correlation coefficients in which the data is replaced by ranks. Both give values from –1 to 1. The closer these values are to 1 the less scattered the data is about a straight line, until at the value of 1 all the points are on a straight line with a positive slope. The same applies for –1 except all the data points lie on a straight line with a negative slope.

For the comparison of precision between ANN and regression the F-test was conducted on the percent relative deviation (signed MRE) data sets.

There is the possibility of bias in the models. This can be tested by an examination of the signed residuals (observed minus predicted). A model is biased if most of its predicted values are significantly below or above the observed values. The null hypothesis that was tested is that the average of residuals is not different from zero.

Sometimes one model is better in a certain response domain than the other. For example, it could be that ANN predicts better for high values of dissolution rate. This can be seen in a scatter plot of the difference between the absolute residuals (absolute value of the error term: observed - predicted) of regression minus those of ANN on the y-axis versus the observed value on the x-axis. Another scatter plot used was of the MRE versus the response value. Two sets of points (one for ANN and one for regression) are plotted in a figure relating to one response. These plots give a good indication of dependency of error upon response values. For both types of scatter plots the data trend lines were plotted.

4.3 Results & discussion

A large number of ANN were trained and the best model for each response was selected for comparison with the corresponding best regression model. Extensive comparison between ANN and regression was done, involving various statistical techniques. The topics covered by these tests were: which method has lower prediction error (for which t-test and bivariate correlation were employed); which method's predictions are more precise; which method is less biased; whether there is a connection between the predictive ability and response value for both ANN and regression.

4.3.1 Summary of the best ANN models

Table 4.1 presents the MRE of the best ANN predictive models. The MRE of the best regression models were presented in the previous chapter. The first column represents the number of neurons in the hidden layer for the backpropagation ANN (columns 2-6) for RBF ANN it represents spread constant interval (which is explained later in this section). The upper row of headings from columns two to nine represents the response variable and the bottom one the training method. All the data in each column (9 values) are relevant for the specific training method. Some ANN parameters were also manipulated for each training method and the MRE of the best one chosen for each column is in the table, i.e. the optimised number of epochs for the basic gradient descent ANN was 10,000, and for the other backpropagation ANN was 1000. For the radial basis function ANN the mean-squared error goal for the responses of impact friability, disintegration time and dissolution rate was 0.1333, 0.1330 and 0.1320 respectively. The spread constant for these responses was 0.915, 0.72 and 0.555 respectively. For the radial basis function ANN the left column represents number of interval (t) of spread constant (SC), e.g. SC = 0.81 + 0.015*t, since the lowest MRE was with t = 7 for the impact friability response the calculated spread constant is 0.915. The equations of the spread constant for disintegration time and dissolution rate are SC = 0.675 + 0.015*t and SC = 0.54 + 0.015*t, respectively. Hence, table 4.1 also demonstrates the sensitivity of radial basis function upon changing the spread constant (only sample data near the optimum spread constant is given). An example of how to read the data from the second column, for the mean weight response the best ANN model was simple backpropagation with basic gradient descent. The best topology had 5 neurons in the hidden layer.

Table 4.1: Average percentage deviation (MRE) between observed and predicted values of ANN. The number in bold represents the best ANN model. The left column represents the number of hidden neurons for the backpropagation training methods (columns 2-6) and for the RB ANN it represents the number of spread constant intervals of 0.015 (the base level differs for each response in columns 7-9). The headings in the upper row from columns 2 to 9 stand for the responses of mean weight, thickness, tensile strength, erosion friability, impact friability, disintegration time and dissolution rate respectively. The bottom row represents the best learning algorithm to which all the data in the columns belong.

No. Hidden Neurons (Backpropagation) / No. of Spread Constant Intervals of 0.015 (RB)	Weight	Thickn.	Hard.	Tensile	Er. Fr.	Im. Fr.	Disint.	Dissol.
1	1.94	1.76	12.62	15.25	10.46	6.86	11.50	**17.73**
2	1.65	1.68	11.87	15.15	11.61	7.08	14.15	19.91
3	1.43	1.37	9.19	12.57	9.40	7.12	**10.50**	24.36
4	1.54	1.25	8.43	10.64	10.20	6.64	12.48	29.85
5	**1.29**	**1.02**	10.68	13.77	9.77	5.81	12.88	27.24
6	1.84	1.18	8.54	9.86	8.55	6.17	15.68	21.78
7	1.80	1.54	28.09	31.75	8.80	**5.75**	18.66	23.83
8	1.71	1.40	**7.39**	**8.11**	**6.57**	7.07	19.61	25.87
9	2.29	1.39	16.41	19.47	8.39	6.94	19.90	25.82
Best Learning Algorithm	BGD	BR	BFGS	BFGS	PRCG	RB	RB	RB

Note. BGD = Basic Gradient Descent
BR = Bayesian Regularization
BFGS = The Quasi-Newton method of Broyden. Fletcher, Goldfarb and Shanno
PRCG = Polak-Ribiere conjugate gradient algorithm
RB = Radial Basis function

Table 4.1 shows that it is worth using training methods other than the simple backpropagation, since the lowest values of MRE were achieved by different training methods. Radial basis function ANN gave the best predictions in 3 out of the 8 responses. The latter result is added to the fact that training with radial basis function is very fast and so is worth trying. The last step in optimising the best radial basis function ANN was to optimise the error goal. This is discussed in section 4.3.1.1.

Tables 4.2 and 4.3 show the percent relative deviation of the best ANN and regression sets respectively. One can see in Table 4.1 that the best MRE dissolution value is 17.73, this value is the mean (of absolute percentage deviation) of 27 cases presented in Table 4.2. Summing up the absolute values of the percent relative deviation presented in Table 4.2, and dividing by 27 will yield this average value of 17.73. These values are generated by computing [(observed-predicted)/observed] × 100. The data presented here (in Tables 4.2

& 4.3) are the signed relative errors from the relevant models (the best ones) used as raw data for the statistical tests. Trying to model dissolution with only one output neuron did not result in a smaller MRE. It was decided hence, not to model other responses with only one output neuron. This decision also took into account the limited computation time. The 8 responses each has 27 cases and in Tables 4.2 and 4.3 are presented respectively as: mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate. From section 4.3.2 onwards a comprehensive statistical comparison of ANN and regression prediction results is discussed.

Table 4.2: Percent relative deviation of the best ANN (abbreviated as "A_") for each case when excluded from training set. The abbreviations for the responses stand for (from second left to right column): mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate respectively.

Case	A_Weight	A_Thickn	A_Hardn	A_Tensil	A_Er_Fri	A_Im_Fri	A_Disint	A_Dissol
1	1.70	0.58	-12.44	-11.35	-0.88	3.86	5.33	25.14
2	-1.13	0.45	4.87	-1.32	-11.17	-2.16	10.47	15.61
3	-0.78	0.18	2.30	2.51	6.82	-3.33	-27.21	-4.53
4	-0.26	0.62	-14.56	-20.05	10.98	5.67	-5.34	-17.46
5	1.02	1.45	1.26	5.13	-11.34	-10.00	7.89	-18.88
6	-4.55	-1.97	4.01	15.21	1.52	3.71	19.77	-79.25
7	0.93	-0.31	7.76	-1.68	0.97	0.47	12.78	1.27
8	-1.98	-1.91	4.66	7.49	-12.66	-5.12	6.84	-77.20
9	0.50	0.28	1.85	-2.03	-3.50	4.19	-19.31	-4.22
10	0.37	0.90	-10.67	-8.15	5.81	6.97	6.31	10.58
11	1.96	-0.08	-3.91	-3.73	-4.81	-4.64	-18.78	48.39
12	1.10	-0.21	3.14	0.92	10.75	-19.96	17.56	-8.10
13	-3.89	1.14	-3.79	-8.91	7.61	-4.64	8.08	-12.46
14	0.37	-0.63	10.92	12.02	8.80	0.66	20.32	15.92
15	-1.21	-0.17	-23.48	-23.23	-4.34	-3.97	-3.88	10.52
16	1.83	2.76	-4.81	-9.80	2.46	3.78	7.11	-1.92
17	-1.26	-0.43	8.97	11.26	14.74	-3.83	17.07	-31.13
18	-1.90	-2.59	-0.31	0.63	-0.56	5.81	2.66	-0.47
19	0.37	0.24	-51.49	-51.80	-1.32	10.37	-7.35	10.40
20	-2.49	-3.02	2.45	8.34	7.07	12.71	-2.13	29.81
21	1.51	3.23	-3.04	-4.36	-12.77	-1.28	10.88	9.53
22	-0.31	0.52	5.08	3.83	-7.80	3.87	-11.62	-3.66
23	0.34	-0.39	-3.91	-1.97	-0.83	11.90	-2.57	-0.38
24	-0.15	-0.14	-3.36	-0.88	10.07	2.44	-22.28	6.67
25	0.11	-1.61	-1.21	-0.74	-6.74	11.45	-0.89	6.94
26	1.65	0.23	0.92	0.75	0.04	6.28	8.83	-2.36
27	1.16	1.54	4.34	-0.88	-11.03	2.16	0.36	25.89

Table 4.3: Percent relative deviation sets of the best regression models (abbreviated as "R_") for each case when excluded from training set. The abbreviations for the responses stand for (from second left to right column): mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate respectively.

Case	R_Weight	R_Thickn	R_Hardnn	R_Tensil	R_Er_Fri	R_Im_Fri	R_Disint	R_Dissol
1	1.05	0.38	-1.81	-9.50	-5.14	0.72	-6.83	34.75
2	-1.02	-0.82	8.60	0.50	-8.98	-6.58	11.50	-5.10
3	-0.16	-0.70	11.56	2.83	6.00	-3.02	-12.77	10.01
4	0.30	1.13	-19.08	-4.54	8.59	6.3	2.21	-19.56
5	1.46	1.18	6.59	16.37	-2.60	-5.07	8.79	-13.49
6	-2.23	-1.89	-10.40	-9.79	-5.13	-2.78	2.59	-93.82
7	1.75	0.16	6.96	2.50	-5.19	-1.92	25.43	15.22
8	-0.97	-1.23	0.26	-2.69	3.28	-6.73	-12.96	-85.95
9	0.13	0.18	5.81	0.15	8.31	-2.03	0.40	2.38
10	-0.31	-0.92	-4.30	-11.38	-11.68	13.49	16.34	-8.87
11	1.62	1.24	-11.40	-0.02	2.40	-6.7	10.63	34.74
12	0.71	0.24	0.33	9.18	4.14	-9.35	-9.25	-22.68
13	-5.43	-0.22	3.27	-1.34	2.55	-0.87	13.28	-22.95
14	0.02	0.46	8.97	4.41	-27.44	9.35	-5.39	16.61
15	-1.09	-0.54	6.30	0.64	8.74	-5.14	-3.87	22.28
16	1.58	2.14	10.96	0.95	14.14	10.13	5.79	5.19
17	-0.65	0.13	-2.48	11.90	3.32	-6.53	6.34	-22.22
18	-1.76	-1.09	-2.40	10.31	4.01	-6.97	-1.43	4.53
19	-0.40	-0.06	-52.33	-47.43	-7.29	-2.58	-3.63	-6.61
20	-2.90	-2.80	-2.42	-1.97	13.69	11.74	-5.26	-20.16
21	1.45	1.48	6.55	-0.54	-13.81	-7.47	10.58	16.12
22	0.56	-0.17	12.74	6.87	-2.14	4.27	-5.23	6.36
23	0.56	0.15	-1.87	8.05	1.55	9.17	-11.61	-0.71
24	0.76	-0.31	-2.79	-6.77	7.61	0.82	-27.01	6.96
25	-0.73	-1.27	-1.65	10.10	-7.18	7.67	-10.09	3.14
26	1.74	0.54	2.68	-1.88	3.08	3.37	-12.29	-6.17
27	3.16	2.29	-1.96	-5.99	-7.63	-1.51	4.85	13.88

4.3.1.1 Arriving at the best ANN that models the dissolution rate response

The mean squared error (MSE) is the value that the ANN tried to minimise. The performance goal parameter (also called error goal) in the ANN is the value of the MSE that upon ANN attaining that value the training will stop. Figure 4.1 demonstrates optimisation of the performance goal value for better prediction of the dissolution response. On the y-axis is the mean relative error value and on the x-axis is the performance goal parameter value (MSE). The MRE represents the quality of prediction since it is a measurment of the predictive ability (since it is calculated on the validation set) whereas the MSE is a quantity that is minimised during training.

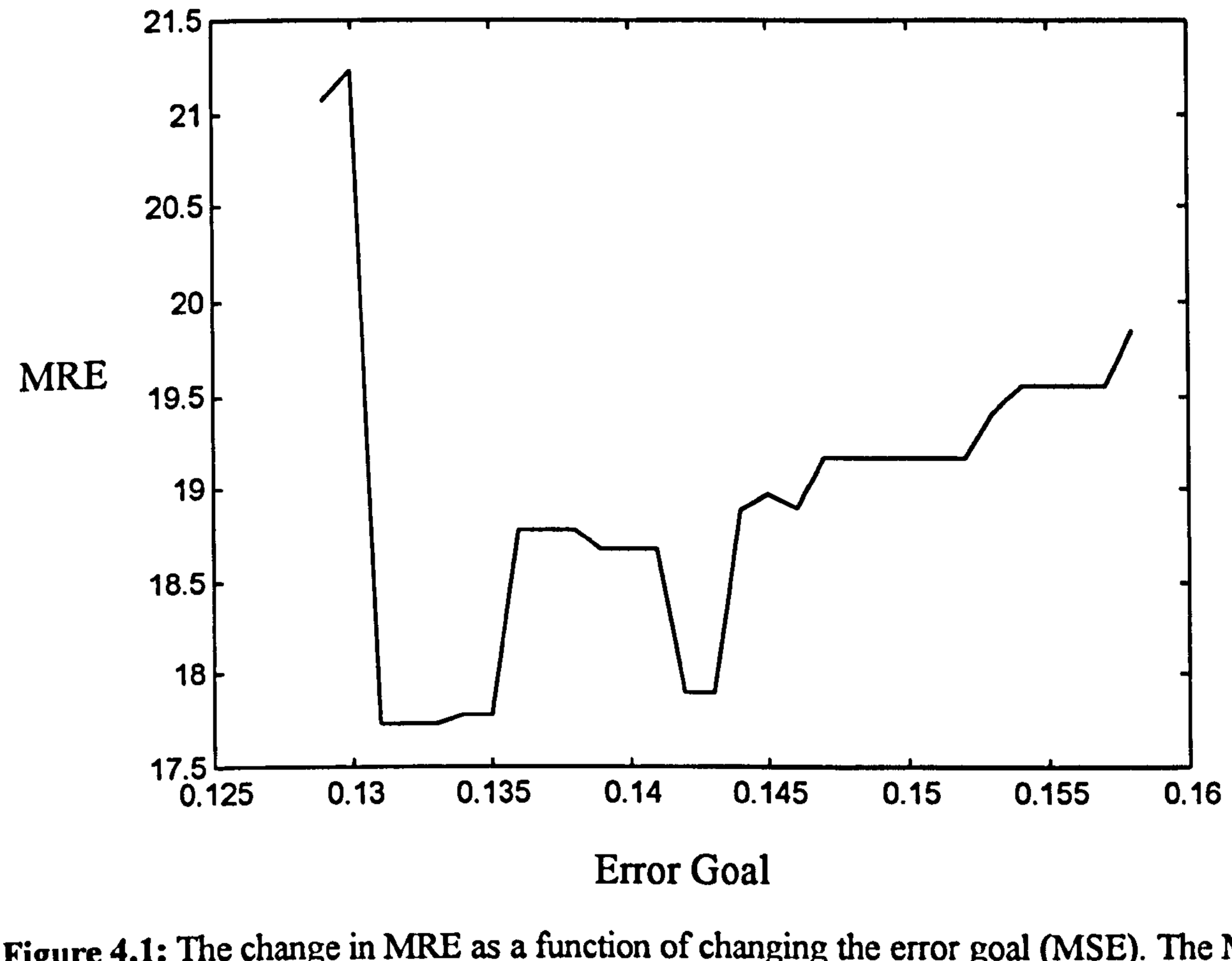


Figure 4.1: The change in MRE as a function of changing the error goal (MSE). The MRE is for the dissolution rate response modeled by radial basis function ANN.

As can be seen in Figure 4.1 when the error goal falls below 0.1333, which was the optimised solution with the minimum MRE for dissolution response, there is a sharp increase in MRE. As the error goal increases there is a more moderate deterioration in the ANN predictive ability. It is hence worth monitoring the radial basis function ANN error goal since it can be quite sensitive to this parameter.

4.3.2 Testing MRE of ANN versus regression

4.3.2.1 Data distribution of relative errors differences

In order to select the appropriate statistical test, for comparison between the two methods, it is necessary to examine the data with respect to the type of population distribution it has. Specifically if there is a normal distribution it is possible to conduct a paired t-test. The initials "DIF" on Figure 4.2 and Table 4.4 represents the difference between ANN and regression absolute percent deviations from observed values. They both present the results of the following responses, mean weight, thickness, hardness, tensile strength, erosion friability, impact friability, disintegration time and dissolution rate respectively (from top figure to bottom one and from left to right column of figures). The y-axis on Figure 4.2 represents the frequency and the x-axis the difference between the absolute percent deviations of ANN and regression for each response. Table 4.4 presents Shapiro-Wilk test that is quantitative statistical test for normality. The significance values are for the null hypothesis that the data are a sample from a normal distribution.

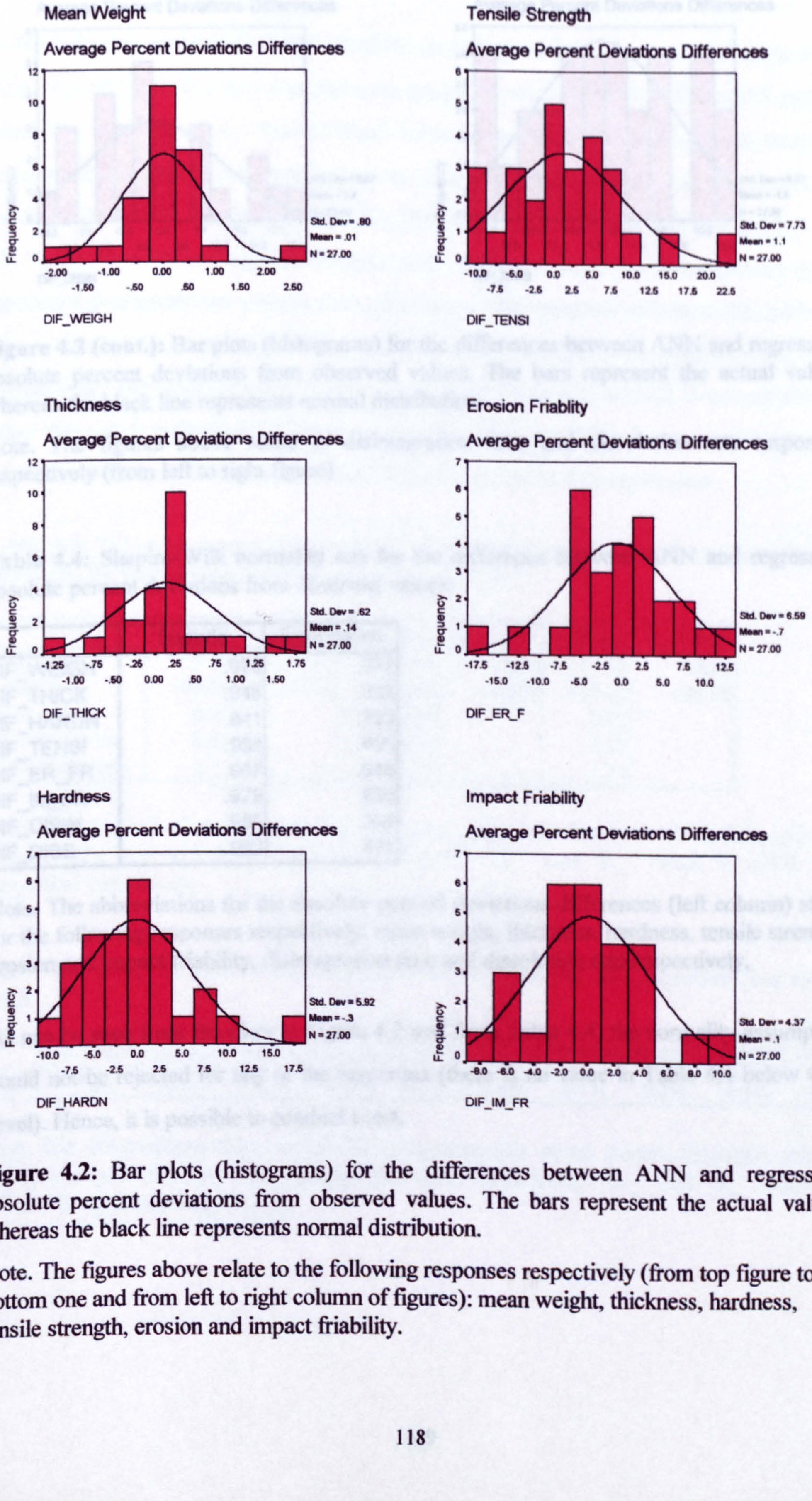


Figure 4.2: Bar plots (histograms) for the differences between ANN and regression absolute percent deviations from observed values. The bars represent the actual values whereas the black line represents normal distribution.

Note. The figures above relate to the following responses respectively (from top figure to bottom one and from left to right column of figures): mean weight, thickness, hardness, tensile strength, erosion and impact friability.

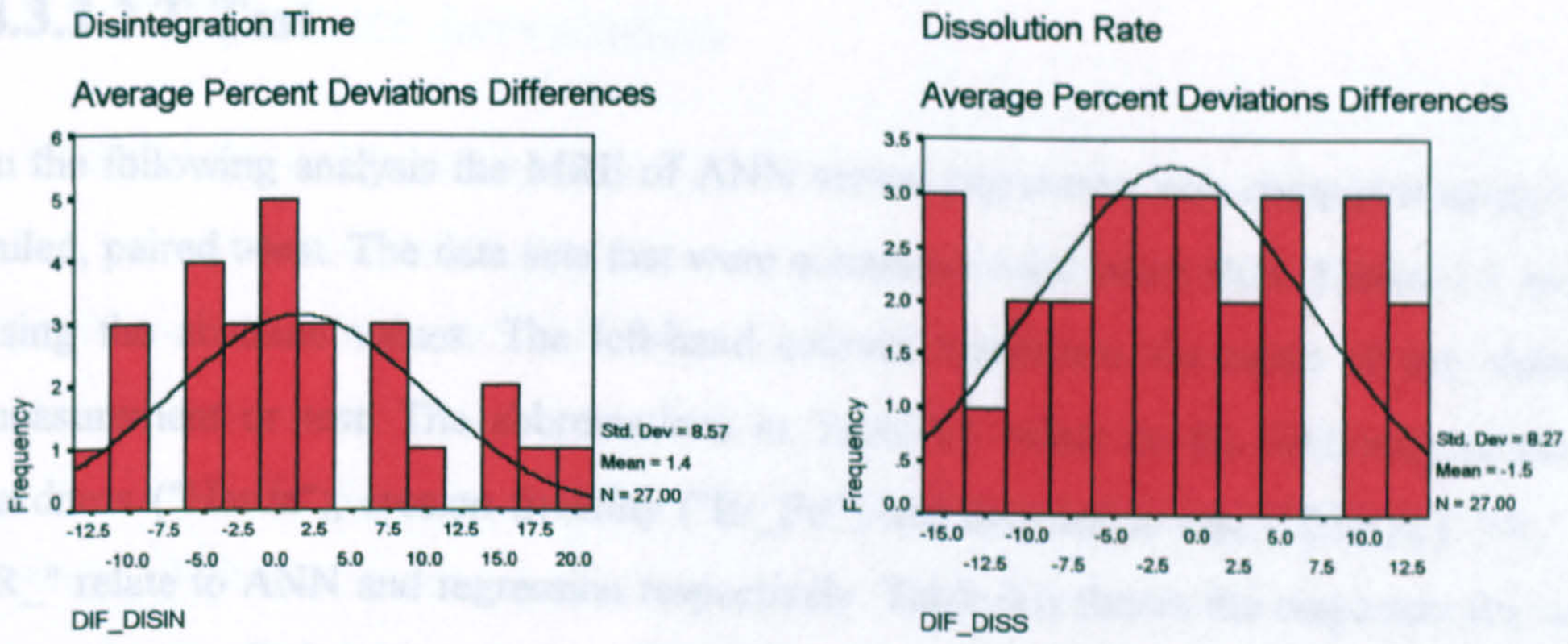


Figure 4.2 (cont.): Bar plots (histograms) for the differences between ANN and regression absolute percent deviations from observed values. The bars represent the actual values whereas the black line represents normal distribution.

Note. The figures above relate to disintegration time and dissolution rate responses respectively (from left to right figure)

Table 4.4: Shapiro-Wilk normality test for the difference between ANN and regression absolute percent deviations from observed values.

	Statistic	Significance
DIF_WEIGH	.928	.072
DIF_THICK	.945	.232
DIF_HARDN	.941	.183
DIF_TENSI	.958	.405
DIF_ER_FR	.967	.545
DIF_IM_FR	.979	.835
DIF_DISIN	.955	.364
DIF_DISS	.963	.471

Note. The abbreviations for the absolute percent deviations differences (left column) stand for the following responses respectively: mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate respectively.

As can be seen from the plots in Figure 4.2 and from Table 4.4, the normality assumption could not be rejected for any of the responses (there is no value in Table 4.4 below 0.05 level). Hence, it is possible to conduct t-test.

4.3.2.2 T-Test

In the following analysis the MRE of ANN versus regression was compared using a one tailed, paired t-test. The data sets that were compared were taken from Tables 4.2 and 4.3 using the absolute values. The left-hand column represents the name of the statistical measurement or test. The abbreviations in Table 4.5 relate to the following responses: hardness ("Hardn"), erosion friability ("Er_Fri") and dissolution rate ("Dissol"). "A_" and "R_" relate to ANN and regression respectively. Table 4.6 shows the responses for which regression predictions were better than ANN ones. The responses shown in this table are mean weight, thickness, tensile strength, impact friability and disintegration time respectively. The t-test statistics is not shown (as in Table 4.5) since it is not possible to reject the null hypothesis with responses that have lower MRE of regression relative to ANN, since the hypothesis tested is that ANN predicts better than regression.

Table 4.5: T-Test to inspect whether MRE of ANN is significantly different from that of regression for the responses with lower MRE for ANN.

	A_Hardn	R_Hardn	A_Er_Fri	R_Er_Fri	A_Dissol	R_Dissol
MRE	7.39	7.65	6.57	7.25	17.73	19.28
Variance	103.54	100.61	20.25	29.89	431.23	498.92
t Stat	-0.23		-0.53		-0.97	
P(T<=t) one-tail	0.41		0.30		0.17	
t Critical one-tail	1.71		1.71		1.71	

Note. The abbreviations relate to the following responses: hardness ("Hardn"), erosion friability (Er_Fri) and dissolution rate ("Dissol"). "A_" and "R_" relate to ANN and regression respectively.

Table 4.6: MRE of ANN versus regression in the responses that regression has lower MRE.

	A_Weight	R_Weight	A_Thickn	R_Thickn	A_Tensil	R_Tensil	A_Im_Fri	R_Im_Fri	A_Disint	R_Disint
MRE	1.29	1.28	1.02	0.88	8.11	6.99	5.75	5.64	10.50	9.13

Note. The abbreviations relate to the following responses: mean weight, thickness, tensile strength, impact friability and disintegration time respectively. "A_" and "R_" relate to ANN and regression respectively.

4.3.3 Bivariate correlation

To determine which method predicts better, and if there is any corrleation between ANN and regression predictions, bivariate correlation was employed. Pearson correlations and Spearman's non-parametric correlations between the responses and the predicted responses of ANN and regression are shown in Tables 4.7a and 4.7b respectively. Tables 4.7a and 4.7b relate to the following responses respectively (from top to bottom in each table): mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate. The headings "Dissolution", "A_Dissolution" and "R_Dissolution" stand for the observed values, the predicted values by ANN and the predicted values by regression for the dissolution response respectively. To read the data, for example, looking at the first table (Table 4.7a) in the upper left-hand corner, on the second row and second column there is a value of 0.939. That means, the Pearson correlation coefficient between ANN predictions and the observed values (for the mean weight response) is 0.939.

Table 4.7a: Pearson correlations between the responses and the predicted responses of ANN and regression.

	WEIGHT	A_WEIGHT	R WEIGHT
WEIGHT	1.000	.939**	0.379*
A_WEIGHT	.939**	1.000	.625**
R WEIGHT	.379*	.625**	1.000
	THICKNESS	A_THICKN	R_THICKN
THICKNESS	1.000	.988**	.947**
A_THICKN	.988**	1.000	.982**
R_THICKN	.947**	.982**	1.000
	HARDNESS	A_HARDN	R_HARDN
HARDNESS	1.000	.967**	.877**
A_HARDN	.967**	1.000	.956**
R_HARDN	.877**	.956**	1.000
	TENSILE	A_TENSIL	R_TENSIL
TENSILE	1.000	.974**	.885**
A_TENSIL	.974**	1.000	.957**
R_TENSIL	.885**	.957**	1.000
	ER_FRIAB	A_ER_FRI	R_ER_FRI
ER_FRIAB	1.000	.988**	.957**
A_ER_FRI	.988**	1.000	.984**
R_ER_FRI	.957**	.984**	1.000
	IM_FRIAB	A_IM_FRI	R_IM_FRI
IM_FRIAB	1.000	.986**	.994**
A_IM_FRI	.986**	1.000	.993**
R_IM_FRI	.994**	.993**	1.000
	DISINTEGRATION	A_DISINT	R_DISINT
DISINTEGRATION	1.000	.975**	.897**
A_DISINT	.975**	1.000	.957**
R_DISINT	.897**	.957**	1.000
	DISSOLUTION	A_DISSOL	R_DISSOL
DISSOLUTION	1.000	.916**	.539**
A_DISSOL	.916**	1.000	.787**
R_DISSOL	.539**	.787**	1.000

*. Correlation is significant at the .05 level (1-tailed).

**. Correlation is significant at the .01 level (1-tailed).

Note. The tables above relate to the following responses respectively: mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate, e.g. "Dissolution", "A_Dissolution" and "R_Dissolution" stand for the observed values, the predicted values by ANN and the predicted values by regression for the dissolution response respectively.

Table 4.7b: Spearman's non-parametric correlations between the responses and the predicted responses of ANN and regression.

	WEIGHT	A_WEIGHT	R WEIGHT
WEIGHT	1.000	.948**	.447**
A_WEIGHT	.948**	1.000	.648**
R WEIGHT	.447**	.648**	1.000
	THICKNESS	A_THICKN	R_THICKN
THICKNESS	1.000	.984**	.915**
A_THICKN	.984**	1.000	.959**
R_THICKN	.915**	.959**	1.000
	HARDNESS	A_HARDN	R_HARDN
HARDNESS	1.000	.946**	.857**
A_HARDN	.946**	1.000	.960**
R_HARDN	.857**	.960**	1.000
	TENSILE	A_TENSIL	R_TENSIL
TENSILE	1.000	.971**	.885**
A_TENSIL	.971**	1.000	.962**
R_TENSIL	.891**	.962**	1.000
	ER_FRIAB	A_ER_FRI	R_ER_FRI
ER_FRIAB	1.000	.979**	.925**
A_ER_FRI	.979**	1.000	.950**
R_ER_FRI	.925**	.950**	1.000
	IM_FRIAB	A_IM_FRI	R_IM_FRI
IM_FRIAB	1.000	.962**	.964**
A_IM_FRI	.962**	1.000	.980**
R_IM_FRI	.964**	.980**	1.000
	DISINTEGRATION	A_DISINT	R_DISINT
DISINTEGRATION	1.000	.952**	.890**
A_DISINT	.952**	1.000	.961**
R_DISINT	.890**	.961**	1.000
	DISSOLUTION	A_DISSOL	R_DISSOL
DISSOLUTION	1.000	.898**	.447**
A_DISSOL	.898**	1.000	.660**
R_DISSOL	.447**	.660**	1.000

**. Correlation is significant at the .01 level (1-tailed).

Note. The tables above relate to the following responses respectively: mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate. e.g. "Dissolution", "A_Dissolution" and "R_Dissolution" stand for the observed values, the predicted values by ANN and the predicted values by regression for the dissolution response respectively.

General examination of the results in the correlation tests between ANN and regression to the actual observed values reveals the superiority of ANN in all the tests. However, both ANN and regression produced models that are valid at the 99% confidence level for all response variables (one exception is the Pearson correlation of regression predicted values to the mean weight observed values that shows only 95% confidence level). The gap between the correlation coefficients of ANN and regression is quite big in the responses of mean weight and dissolution rate.

4.3.4 Precision test of ANN versus regression

To compare the precision of ANN versus regression, F-Tests for variances of percent relative errors (signed MRE) between ANN and regression were conducted, and the results are presented in Table 4.8. The left-hand column is the name of the statistical measurement/test involved. The other columns are the values of these measurements/tests. The abbreviations for the responses are (from left to right column and continued below): mean weight, thickness, hardness, tensile strength, erosion friability, impact friability, disintegration time and dissolution rate respectively. "A_" and "R_" stand for ANN and regression percent relative deviation respectively.

Table 4.8: F-Test for variances of percent relative errors between ANN and regression.

	A_Weight	R_Weight	A_Thickn	R_Thickn	A_Hardn	R_Hardn	A_Tensil	R_Tensil
Mean	-0.19	-0.03	0.03	-0.01	-2.76	-0.86	-3.07	-0.71
Variance	2.85	3.03	2.03	1.35	152.34	160.57	171.81	135.38
F	0.94		1.50		0.95		1.27	

	A_Er_Fri	R_Er_Fri	A_Im_Fri	R_Im_Fri	A_Disint	R_Disint	A_Dissol	R_Dissol
Mean	-0.08	-0.47	1.38	-0.06	1.52	-0.33	-1.68	-5.04
Variance	65.07	84.18	51.79	45.28	166.07	127.92	754.70	858.41
F	0.77		1.14		1.30		0.88	

Note. The abbreviations for the responses stands for (from left to right column and from top table to bottom one): mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate respectively. "A_" and "R_" stand for initials that relate to ANN and regression percent relative deviation respectively.

In the precision test data presented in Table 4.8 one can see at the thickness response that the variance of regression relative errors is 1.35 whereas the variance of ANN relative errors is 2.03 and the calculated F value is 1.50. Hence, the regression model for the thickness response is about 50% more precise than the ANN one ("about" because F-test is done on idependent samples and not paired data as in this study, so it is not possible to use significance values generated in this test).

4.3.5 Bias examination

To examine if the predicted values generated by ANN and regression models are significantly below or above the observed values, bias tests were conducted. Bias tests on signed residuals (observed minus predicted) of ANN ("A_") and regression ("R_") are shown in Table 4.9. The left-hand column gives information on whether the model is ANN or regression one and the response modelled. The abbreviations in Table 4.9 from top downwards stand for responses of: disintegration time, dissolution rate, erosion friability, hardness, impact friability, tensile strength, thickness and mean weight respectively. This set of abbreviations is repeated since it relates to ANN as well as regression.

Table 4.9: Bias test on signed residuals (observed minus predicted) of ANN ("A_") and regression ("R_").

One-Sample Test

	Test Value = 0					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
A_DISINT	.794	26	.434	1.060E-02	-1.69E-02	3.805E-02
A_DISSOL	.471	26	.641	9.595E-03	-3.22E-02	5.144E-02
A_ER_FRI	.335	26	.740	2.611E-03	-1.34E-02	1.863E-02
A_HARDNN	-.993	26	.330	-1.41E-02	-4.32E-02	1.507E-02
A_IM_FRI	1.061	26	.298	6.725E-03	-6.30E-03	1.975E-02
A_TENSIL	-1.027	26	.314	-1.48E-02	-4.44E-02	1.482E-02
A_THICKN	.160	26	.874	3.400E-04	-4.03E-03	4.707E-03
A_WEIGHT	-.508	26	.616	-1.41E-03	-7.13E-03	4.304E-03
R_DISINT	.084	26	.934	9.02E-04	-2.1E-02	2.29E-02
R_DISSOL	-.210	26	.835	-4.21E-03	-4.5E-02	3.70E-02
R_ER_FRI	-.121	26	.905	-8.32E-04	-1.5E-02	1.33E-02
R_HARDNN	.116	26	.908	1.54E-03	-2.6E-02	2.89E-02
R_IM_FRI	.056	26	.955	3.09E-04	-1.1E-02	1.15E-02
R_TENSIL	-.010	26	.992	-1.15E-04	-2.4E-02	2.34E-02
R_THICKN	-.008	26	.994	-1.37E-05	-3.6E-03	3.62E-03
R_WEIGHT	-.014	26	.989	-3.96E-05	-6.0E-03	5.89E-03

Note. The abbreviations from top downwards stand for responses of: disintegration time, dissolution rate, erosion friability, hardness, impact friability, tensile strength, thickness and mean weight respectively. This set of abbreviations is repeated twice since it relates to ANN as well as regression.

Looking in Table 4.9, one can see that there are high significance values for all the tests. It can be seen from the bias of ANN and regression models, that there is no one model that violates the t-test (the null hypothesis is that the mean values of signed residuals is not differ from zero), so there is no bias in any of the models. One can also examine the 95% confidence interval and see that the zero value is within the interval for all the models.

4.3.6 The predictive ability as a function of the response value

To examine the influence of the response value on the predictive ability of ANN versus regression, two different sets of plots were employed and will be presented in this subsection.

Figure 4.3 shows the difference between the absolute residuals (observed-predicted) of regression minus those of ANN on the y-axis, as a function of the observed value on the x-axis (scaled values). The initials "RD" on the y-axis stands for the residual difference. The plots of Figure 4.3 relate to the following responses (from top figure to bottom one and from left to right column of figures): mean weight, thickness, hardness, tensile strength, erosion and impact friability, disintegration time and dissolution rate respectively.

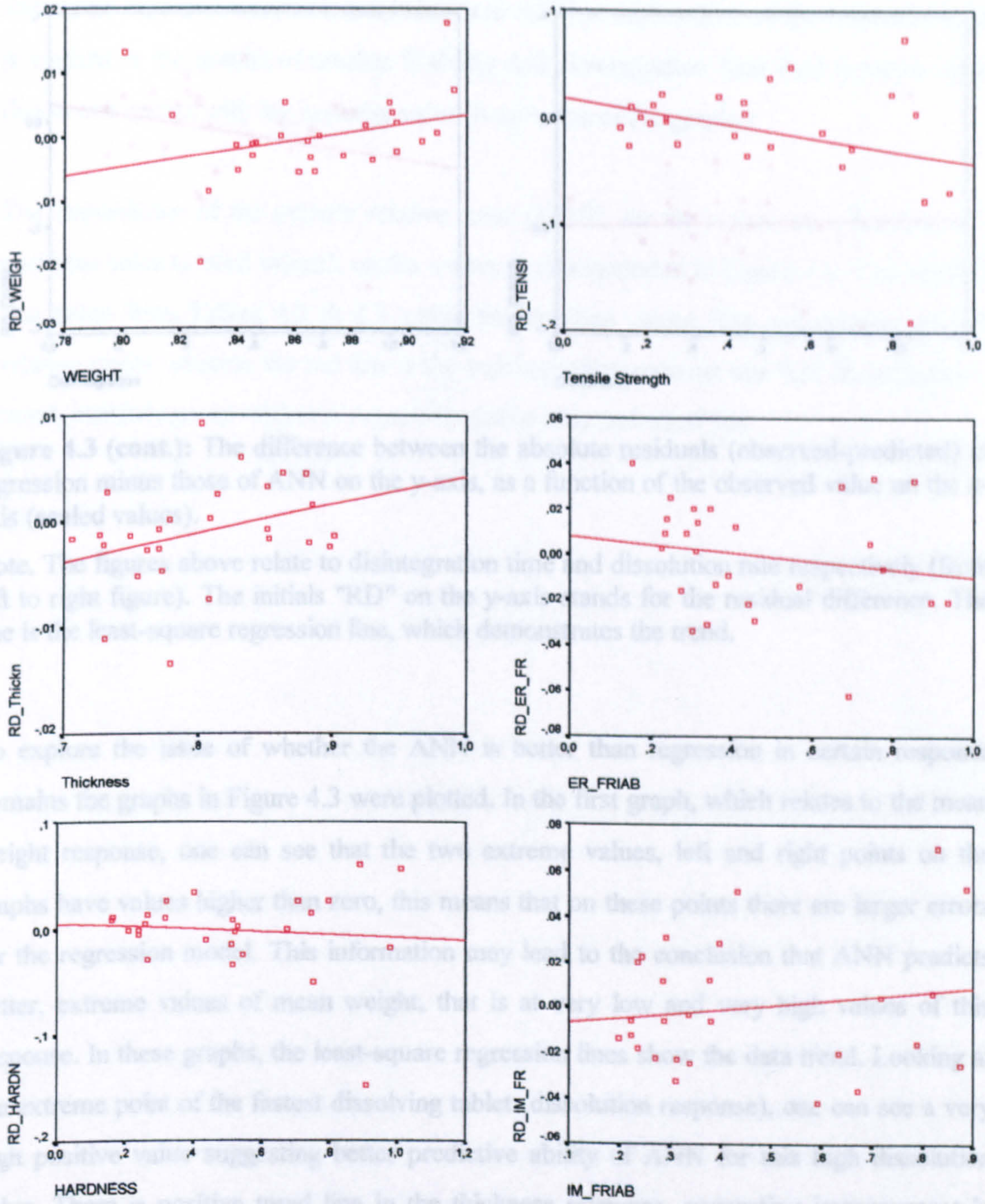


Figure 4.3: The difference between the absolute residuals (observed-predicted) of regression minus those of ANN on the y-axis, as a function of the observed value on the x-axis (scaled values).

Note. The figures above relate to the following responses (from top figure to bottom one and from left to right column of figures) respectively: mean weight, thickness, hardness, tensile strength, erosion and impact friability. The initials "RD" on the y-axis stands for the residual difference. The line is the least-square regression line, which demonstrates the trend.

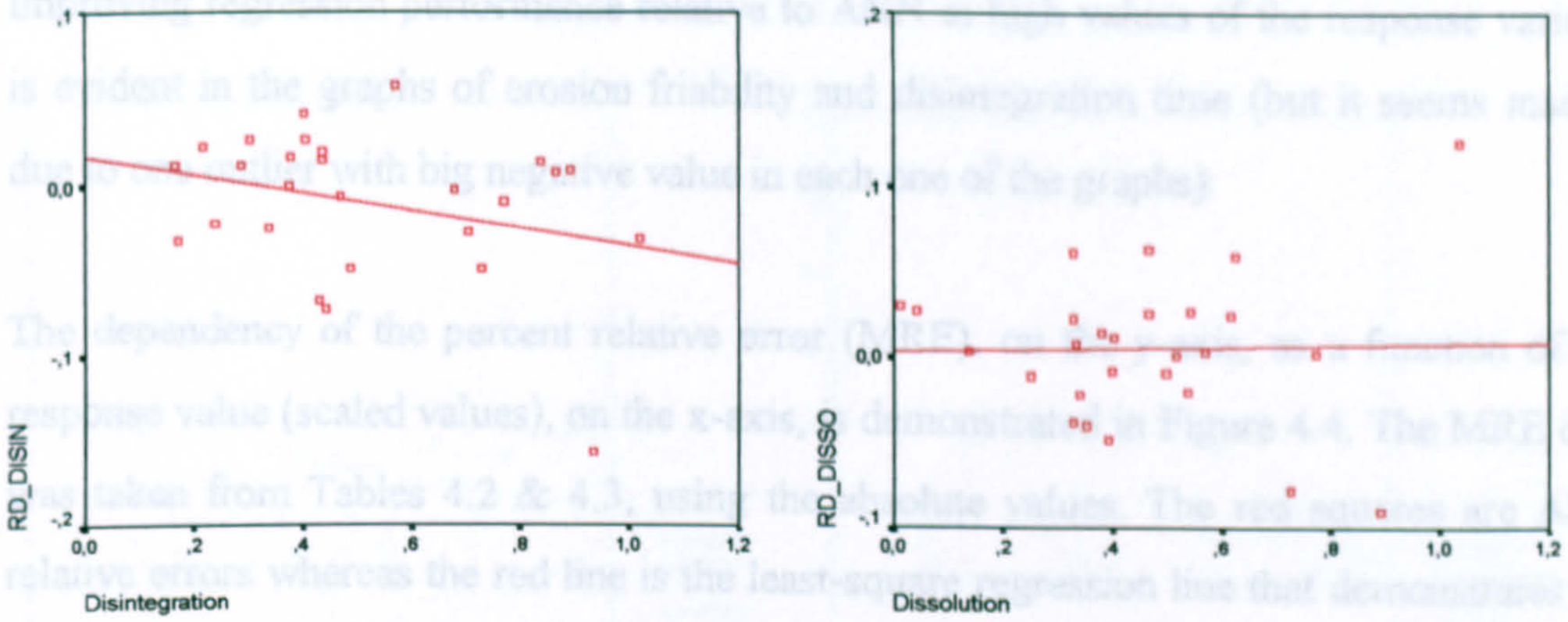


Figure 4.3 (cont.): The difference between the absolute residuals (observed-predicted) of regression minus those of ANN on the y-axis, as a function of the observed value on the x-axis (scaled values).

Note. The figures above relate to disintegration time and dissolution rate respectively (from left to right figure). The initials "RD" on the y-axis stands for the residual difference. The line is the least-square regression line, which demonstrates the trend.

To explore the issue of whether the ANN is better than regression in certain response domains the graphs in Figure 4.3 were plotted. In the first graph, which relates to the mean weight response, one can see that the two extreme values, left and right points on the graphs have values higher than zero, this means that on these points there are larger errors for the regression model. This information may lead to the conclusion that ANN predicts better, extreme values of mean weight, that is at very low and very high values of this response. In these graphs, the least-square regression lines show the data trend. Looking at the extreme point of the fastest dissolving tablet (dissolution response), one can see a very high positive value suggesting better predictive ability of ANN for this high dissolution value. There is positive trend line in the thickness response, suggesting improvement in ANN predictions (as the response became bigger) relative to regression ones and in the impact friability response (in the latter, the slope is mainly due to two outliers at the upper right corner of the graph). The hardness regression trend line suggests ANN performance deteriorates relative to regression as hardness increases, but a close look reveals the slope is quite small and is mainly due to an outlier with a large negative value which belongs to a high hardness value. Observing the tensile strength one can see on the right-hand side of the graph 3 points with values that are considerably less than zero. The latter fact would suggest that the regression model is better at predicting high values of tensile strength and the negative slope of the regression trend line tells the same story. The same trend of

improving regression performance relative to ANN in high values of the response variable is evident in the graphs of erosion friability and disintegration time (but it seems mainly, due to one outlier with big negative value in each one of the graphs).

The dependency of the percent relative error (MRE), on the y-axis, as a function of the response value (scaled values), on the x-axis, is demonstrated in Figure 4.4. The MRE data was taken from Tables 4.2 & 4.3, using the absolute values. The red squares are ANN relative errors whereas the red line is the least-square regression line that demonstrates the trend, similarly green relates to regression data points and trend line.

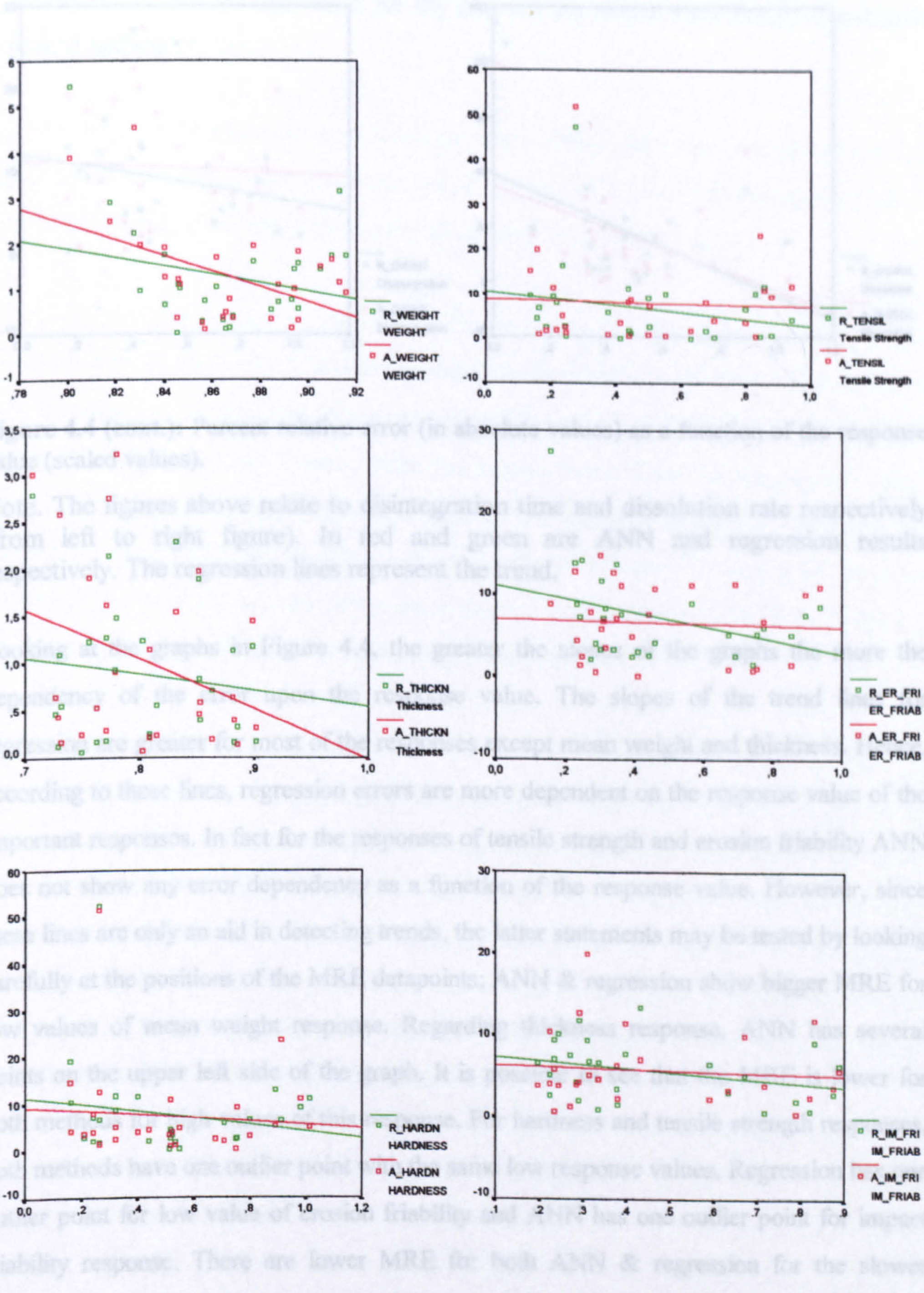


Figure 4.4: Percent relative error (in absolute values) as a function of the response value (scaled values).

Note. The figures above relate to the following responses (from top figure to bottom one and from left to right column of figures) respectively: mean weight, thickness, hardness, tensile strength, erosion and impact friability. In red and green are ANN and regression results respectively. The regression lines represent the trend.

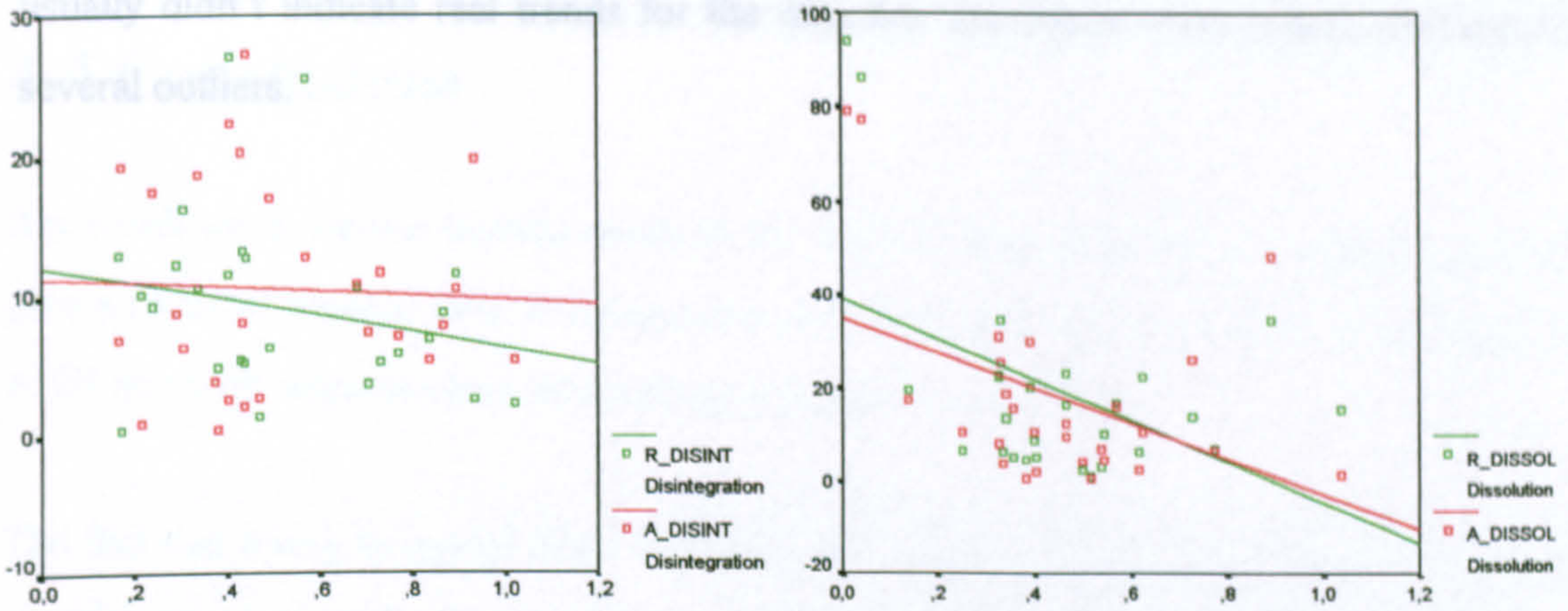


Figure 4.4 (cont.): Percent relative error (in absolute values) as a function of the response value (scaled values).

Note. The figures above relate to disintegration time and dissolution rate respectively (from left to right figure). In red and green are ANN and regression results respectively. The regression lines represent the trend.

Looking at the graphs in Figure 4.4, the greater the slopes of the graphs the more the dependency of the error upon the response value. The slopes of the trend lines for regression are greater for most of the responses except mean weight and thickness. Hence, according to these lines, regression errors are more dependent on the response value of the important responses. In fact for the responses of tensile strength and erosion friability ANN does not show any error dependency as a function of the response value. However, since these lines are only an aid in detecting trends, the latter statements may be tested by looking carefully at the positions of the MRE datapoints; ANN & regression show bigger MRE for low values of mean weight response. Regarding thickness response, ANN has several points on the upper left side of the graph. It is possible to see that the MRE is lower for both methods for high values of this response. For hardness and tensile strength responses, both methods have one outlier point with the same low response values. Regression has one outlier point for low value of erosion friability and ANN has one outlier point for impact friability response. There are lower MRE for both ANN & regression for the slower disintegrating tablets. The big slope of the dissolution rate trend line, for both ANN and regression, could be misleading because it is big due to 2 outliers of the slowest dissolving tablets. The 2 outliers not only cause the greater slope for the 2 trend lines but also considerably alleviate the average MRE for both ANN and regression. The extremely large MRE of the slowest dissolved tablets demonstrates the inability of the models to predict values of very low response values of dissolution rate. To summarise, the trend lines

usually didn't indicate real trends for the data but the slopes were mainly dominated by several outliers.

4.4 Conclusions

It is worth using various training methods for ANN. Using radial basis function ANN could not only give a reduced training time advantage but also could give more accurate predictions. This type of ANN however requires close monitoring of the error goal value.

The fact that trying to model ANN with only one output neuron resulted in a bigger MRE than the complex topology of 8 output neurons reveals that complex topology does not mean a disadvantage. The advantage of training with 8 output neurons lies not just with computation time (because all are computed once) and ease of optimisation but is also an aid in the learning process. The reason for this is that there is information embedded in the weights regarding correlation between the responses, which is lost if one takes a topology of one response for each ANN. It is obvious that disintegration time can provide information about dissolution rate, since there is some correlation between these two responses. This issue will be tackled later.

Bivariate correlations between the predicted values of ANN/regression to the observed values show better results for ANN in the parametric and non-parametric tests for mean weight response. ANN shows lower MRE than regression for the predictions of hardness, erosion friability and dissolution rate responses. T-tests that were conducted on the absolute relative percent deviations (MRE) showed ANN models don't have lower MRE (that are statistically significant). Comparing the MRE results, ANN also predicts better extreme values of the mean weight and dissolution rate responses. Both ANN & regression predict better high values of disintegration time. ANN has more difficulty than regression in predicting low values of thickness.

5. Modelling Properties of Powders using Artificial Neural Networks and Regression: The Case of Limited Data

5.1 Introduction

Artificial neural networks (Hussain et al., 1991) and regression (Schwartz et al., 1973) are commonly used for modelling relationships. This chapter examines the ability of both ANN and regression to model properties of powders when only a limited amount of data is available. The data used here was taken from a published article dealing with hard-gelatin capsule formulations (Hogan et al., 1996). This data is very limited (in terms of characterising space with so many dimensions) as the study involved only 33 experiments although there were 9 independent variables that were manipulated over 4 different levels for each variable. For comparative purposes, a full factorial design conducted at 4 levels would mean 262144 (4^9) experiments and this is clearly not feasible. Comprehensive statistical comparison was used in an attempt to answer the following questions; which method is more accurate; does the accuracy depend on the response value; which method predicts better the extreme response values; which method is more precise; which method is less biased. One question that may be asked why repeat the same work on another data set? The answer is that it is possible when modelling different data types that the answers to the questions would change from the previous study. The type of data in this study is different from the tablet study since it also has qualitative variables (disintegrant type and filler type) and not just quantitative variables. The use of this type of variable for modelling by ANN and regression has been done (Kesavan & Peck, 1995). So these studies and the studies in previous chapters are complementary. It is expected that the ability to model the data using both ANN and regression would be poor due to limited data. The question of what to do with limited data arises in pharmaceutical companies all the time. Hence, they need to make decisions using limited data. Companies face the problem of whether to ignore past experiments (that were not well designed) and to conduct better ones that are well designed. It

is possible that the solution to an optimisation problem lies in the old experiments, and that with good modelling and optimisation techniques this solution will be found. If such a scenario occurs, the work of analysing data could save the time and money of lab work.

5.2 Methods

5.2.1 Experimental data

In the capsule formulation study, (the raw data was taken from Hogan et al., 1996) a total of 33 formulations of hard gelatin capsules were manufactured. The independent variables that were manipulated were (from left to right in Table 5.1): drug particle size (μm), drug solubility (g/l), filler type (the values are ranked data according to the relative solubility of the fillers), filler level (%), disintegrant type (characterised by their relative swelling volume in percent), disintegrant level (%), lubricant level (%), glidant level (%), and drug concentration (%). The drugs used were phenytoin, theophylline, paracetamol, propranolol and aminophylline. The fillers used were ('filler type' relates to the choice of filler) calcium phosphate, microcrystalline cellulose, maize starch, starch 1500, lactose monohydrate. The disintegrants used were ('disintegrant type' relates to the choice of disintegrant) Explotab, AcDiSol, Amberlite, Polyplasdone XL and maize starch. The lubricant and glidant used were magnesium stearate and Aerosil respectively. The response variables were minimum bulk density (gcm^{-3}), maximum bulk density (gcm^{-3}), Hausner's ratio, Carr's compressibility index (%), coefficient of fill weight variation (%), area under the dissolution curve (%min), mean dissolution time (min), variance of the dissolution time and disintegration time. Overall, 9 independent variables and 9 response variables were examined. The data comprising the independent variables and the responses are given in Tables 5.1 and 5.2.

Table 5.1: Independent variables that were manipulated.

Case	D (ps)	D (sol.)	ft	fl	dt	dl	ll	gl	dc
1	26.00	15.00	2	44.00	1680	5.00	1.00	0.00	50.00
2	26.00	15.00	2	43.50	1680	5.00	1.00	0.50	50.00
3	26.00	15.00	2	42.50	1680	5.00	1.00	1.50	50.00
4	26.00	15.00	2	42.00	1680	5.00	1.00	2.00	50.00
5	26.00	15.00	2	44.00	1680	5.00	0.00	1.00	50.00
6	26.00	15.00	2	43.50	1680	5.00	0.50	1.00	50.00
7	26.00	15.00	2	42.50	1680	5.00	1.50	1.00	50.00
8	26.00	15.00	2	42.00	1680	5.00	2.00	1.00	50.00
9	26.00	15.00	2	48.00	1680	0.00	1.00	1.00	50.00
10	26.00	15.00	2	45.50	1680	2.50	1.00	1.00	50.00
11	26.00	15.00	2	40.50	1680	7.50	1.00	1.00	50.00
12	26.00	15.00	2	38.00	1680	10.00	1.00	1.00	50.00
13	65.00	0.20	2	43.00	1680	5.00	1.00	1.00	50.00
14	57.00	8.00	2	43.00	1680	5.00	1.00	1.00	50.00
15	26.00	15.00	2	43.00	1680	5.00	1.00	1.00	50.00
16	122.00	50.00	2	43.00	1680	5.00	1.00	1.00	50.00
17	26.00	200.00	2	43.00	1680	5.00	1.00	1.00	50.00
18	26.00	15.00	2	73.00	1680	5.00	1.00	1.00	20.00
19	26.00	15.00	2	58.00	1680	5.00	1.00	1.00	35.00
20	26.00	15.00	2	28.00	1680	5.00	1.00	1.00	65.00
21	26.00	15.00	2	13.00	1680	5.00	1.00	1.00	80.00
22	26.00	15.00	1	43.00	1680	5.00	1.00	1.00	50.00
23	26.00	15.00	3	43.00	1680	5.00	1.00	1.00	50.00
24	26.00	15.00	4	43.00	1680	5.00	1.00	1.00	50.00
25	26.00	15.00	5	43.00	1680	5.00	1.00	1.00	50.00
26	26.00	15.00	2	43.00	600	5.00	1.00	1.00	50.00
27	26.00	15.00	2	43.00	190	5.00	1.00	1.00	50.00
28	26.00	15.00	2	43.00	150	5.00	1.00	1.00	50.00
29	26.00	15.00	2	43.00	110	5.00	1.00	1.00	50.00
30	26.00	15.00	1	48.00	1680	0.00	1.00	1.00	50.00
31	26.00	15.00	1	38.00	1680	10.00	1.00	1.00	50.00
32	26.00	15.00	5	48.00	1680	0.00	1.00	1.00	50.00
33	26.00	15.00	5	38.00	1680	10.00	1.00	1.00	50.00

Note. The independent variables that were manipulated are: drug particle size (D (ps)), drug solubility (D (sol.)), filler type (ft), filler level (fl), disintegrant type (dt), disintegrant level (dl), lubricant level (ll), glidant level (gl), and drug concentration (dc).

Table 5.2: Responses measured.

Case	Vmin (gcm^{-3})	Vmax (gcm^{-3})	H	Carr (%)	CFV (%)	AUC (%min)	MDT (min)	VDT (min^2)	DT (min)
1	0.50	0.82	1.63	38.79	15.09	2480.0	28.6	107.6	10.5
2	0.50	0.79	1.58	36.71	1.19	1268.0	15.1	23.1	8.6
3	0.46	0.72	1.58	36.81	2.40	957.0	12.5	14.0	8.2
4	0.43	0.68	1.58	36.76	1.28	1907.0	19.0	42.4	7.5
5	0.50	0.76	1.52	34.21	2.51	1994.0	18.0	54.9	6.9
6	0.50	0.72	1.44	30.56	1.86	1733.0	17.4	33.0	7.2
7	0.48	0.71	1.48	32.39	1.36	1869.0	18.8	49.5	8.4
8	0.47	0.70	1.49	32.86	1.18	1775.0	17.3	45.5	10.4
9	0.49	0.72	1.47	31.94	0.95	1069.0	11.0	16.3	9.0
10	0.50	0.74	1.49	33.11	0.79	2087.0	20.8	72.4	8.2
11	0.50	0.71	1.43	30.28	1.44	2719.0	25.5	111.4	7.7
12	0.48	0.74	1.54	34.90	4.06	1834.0	19.7	55.9	7.9
13	0.54	0.78	1.44	30.57	1.40	200000.0	2000.0	200000.0	6.6
14	0.56	0.80	1.43	30.19	0.72	1980.0	19.1	70.6	8.2
15	0.49	0.74	1.51	33.78	0.75	706.0	9.0	7.8	9.9
16	0.62	0.83	1.33	24.70	1.90	2993.0	28.3	128.4	7.6
17	0.53	0.80	1.51	33.75	0.90	1571.0	22.2	41.3	11.5
18	0.55	0.78	1.42	29.49	0.80	1361.0	16.4	31.2	7.5
19	0.52	0.76	1.47	31.79	0.98	1593.0	16.9	45.6	7.7
20	0.44	0.67	1.51	33.58	2.38	2486.0	22.6	98.7	7.4
21	0.40	0.62	1.53	34.68	3.57	2032.0	18.8	74.8	11.4
22	0.54	0.80	1.48	32.50	0.85	2044.0	19.0	68.1	10.8
23	0.46	0.72	1.59	37.24	1.25	3081.0	28.1	147.8	7.4
24	0.40	0.58	1.46	31.30	1.64	1261.0	12.7	27.3	7.0
25	0.38	0.64	1.66	39.84	18.52	2012.0	19.9	56.6	7.6
26	0.44	0.73	1.64	39.04	0.84	1546.0	16.3	39.6	7.7
27	0.46	0.72	1.57	36.11	1.12	3356.0	29.1	172.8	9.3
28	0.44	0.68	1.57	36.50	1.10	3683.0	29.6	229.3	9.8
29	0.46	0.73	1.59	36.99	0.96	1336.0	14.5	29.4	7.6
30	0.54	0.83	1.52	34.34	2.42	7121.0	70.4	840.5	12.1
31	0.53	0.80	1.52	34.16	1.32	2798.0	25.0	117.1	8.9
32	0.40	0.61	1.54	35.25	20.67	82319.0	760.1	117068.5	10.0
33	0.42	0.65	1.57	36.15	5.94	2467.0	25.1	86.3	7.3

Note. The response variables measured (presented from left to right) were: minimum bulk density (Vmin), maximum bulk density (Vmax), Hausner's ratio (H), Carr's compressibility index (Carr), coefficient of fill weight variation (CFV), area under the dissolution curve (AUC), mean dissolution time (MDT), variance of the dissolution time (VDT), and disintegration time (DT) respectively.

5.2.2 Development of ANN and regression models

The ANN models were developed by examining different topologies and different learning methods including backpropagation and radial basis function ANN. There were 9 input units and 9 output neurons in all the topologies. The effect of number of epochs (iterations) was also explored. The regression equations were derived only from a simple linear model of the form $a_1*x_1 + a_2*x_2 + \ldots + a_9*x_9 = y$. Hence the maximum coefficients in the model could only be 9. Four different methods of variable selection were employed: no selection (simple linear model), stepwise regression, backward elimination and forward selection.

In ANN the input and output data was scaled according to zscore and values between 0.1-0.9. A detailed explanation of this scaling method (method 4) is described in Chapter 3 (section 3.2). There was no scaling in the quantitative variables of regression. With regard to the two qualitative variables that relate to the choice of filler and disintegrant dummy variables were used (Mendenhall & Sincich, 1996). They are called dummy variables since the numbers assigned to the various levels are arbitrarily selected. One coding scheme that was tested was to assign one variable to each disintegrant/lubricant. When the disintegrant/lubricant is part of the formulation the related variable gets a value of 1 and if not, a value of 0. Hence, in this study, it means the choice of disintegrant and filler are represented by 10 variables (5 possibilities for each choice of disintegrant/lubricant). This method is advantageous for the interpretation of the regression equation over the method of putting the values as they are in only two dummy variables, a method that was also examined in this study. Obviously, when one tries to predict with regression or ANN he should aware that the input values entered could not include, in this case, other disintegrant/filler than the ones that were used to build the models.

The generalisation ability of both the ANN and regression models was determined using the leave-one-out method (Hussain et al., 1994). In this method all the data is used for training and for validation. A total of 33 data points (cases) are used with 32 of the cases used for building the model and one for validation. By subsequently excluding another case from the model and building the model according to the other 32 cases and performing this procedure for all 33 cases a validation set consisting of 33 cases is generated. The same methodology of comparison between ANN and regression was applied as in the study in the previous chapter, except that Wilcoxon signed-rank test (Norusis, 1997) that is a non-

parametric test was used instead of t-test. The Wilcoxon signed-rank test takes into account the size of the differences between the two related pairs of MRE and gives each one a rank. The smallest difference get rank of value 1, the one afterwards get the value 2 and so on, e.g. 10, 5, –4, 0, get ranks of 3, 2, 1, and the 0 value is ignored. The positive ranks (3 + 2 = 5) and the negative ranks (= 1) are then summed up separately for the comparison. This non-parametric test tests the same null hypothesis as the t-test that was used in Chapter 4. Hence, a one tailed test was conducted to see if ANN predicts significantly better than regression.

An example of one ANN topology used in this study is presented in Figure 5.1. The first layer on the left is the input layer. The middle layer is called the hidden layer since it is not visible to the outside. The first layer on the right is called output layer since the output of the ANN is going out through this layer. The latter two layers are composed of perceptrons since the first layer on the left is just composed of input units that do not apply any operation on their data. This type of ANN is called feedforward, multilayer perceptron (MLP).

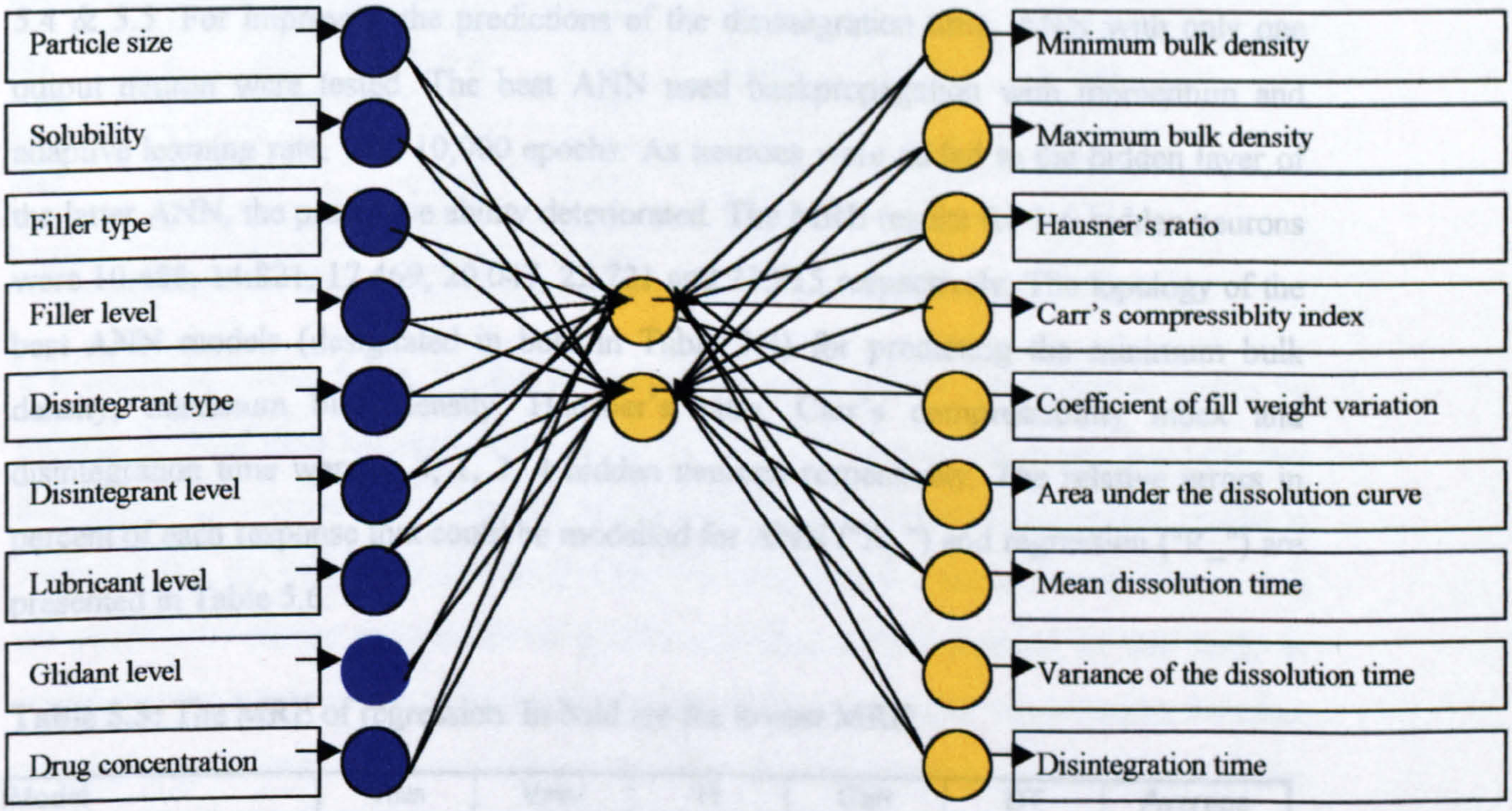


Figure 5.1: Example of one ANN topology used in the study. The input units, which receive data from the independent variables, are on the first layer on the left. The output layer, which outputs the response variables, is the first one to the right. In this case there are only two perceptrons in the hidden layer, but the number of perceptrons in the hidden layer was varied.

5.3 Results & discussion

This study tried to develop ANN and regression models that will succeed to predict all 9 responses of the capsule data using the leave-one-out validation method. Both methods succeeded in predicting just 5 out of the 9 response variables. The responses that were modelled successfully were minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time. The best learning method for the ANN was backpropagation with momentum and adaptive learning rate.

5.3.1 MRE results

All the numbers tabulated in Tables 5.3-5.5 represent the MRE. The results of the leave-one-out for validation are presented in Tables 5.3 & 5.4. ANN that were trained on all 33 cases and tested on these cases (termed 'screening experiments' in previous chapter) are presented in Table 5.5. The results of the best topologies of the ANN are shown in Tables 5.4 & 5.5. For improving the predictions of the disintegration time, ANN with only one output neuron were tested. The best ANN used backpropagation with momentum and adaptive learning rate, with 10,000 epochs. As neurons were added to the hidden layer of the latter ANN, the predictive ability deteriorated. The MRE results for 1-6 hidden neurons were 10.488, 14.821, 17.469, 20.047, 22.721 and 23.815 respectively. The topology of the best ANN models (designated in bold in Table 5.4) for predicting the minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time were 4, 4, 1, 3, 4 hidden neurons respectively. The relative errors in percent of each response that could be modelled for ANN ("A_") and regression ("R_") are presented in Table 5.6.

Table 5.3: The MRE of regression. In bold are the lowest MRE.

Model	Vmin	Vmax	H	Carr	DT	Average
Simple Linear	**4.165**	4.761	3.413	6.205	13.535	6.416
Stepwise Regression	4.238	**4.158**	2.908	5.548	**13.037**	**5.978**
Backward elimination	4.238	4.833	**2.857**	**5.488**	**13.037**	6.091
Forward selection	4.238	**4.158**	2.908	5.548	**13.037**	**5.978**

Table 5.4: The MRE of backpropagation with momentum and adaptive learning rate ANN using the leave-one-out method for validation. In bold are the lowest MRE.

Epochs	Vmin	Vmax	H	Carr	DT	Average
10	11.064	8.369	5.001	9.043	18.997	10.495
20	8.853	7.743	3.992	7.450	15.619	8.731
40	8.798	7.249	3.951	8.197	14.042	8.447
80	7.662	6.432	3.674	6.863	14.465	7.819
160	5.789	5.701	3.250	6.371	13.984	7.019
320	4.580	4.172	3.431	6.853	14.360	6.679
640	4.542	4.037	3.153	5.960	**12.144**	**5.967**
1000	3.944	4.548	3.021	**5.860**	13.007	6.076
1280	3.971	3.649	**2.861**	5.960	14.238	6.136
2000	3.784	**3.548**	3.155	6.235	14.368	6.218
2560	**3.688**	3.680	3.220	6.367	14.225	6.236
10000	4.307	3.724	3.206	6.339	12.297	5.975
20000	3.958	3.863	3.359	6.468	15.522	6.634

Table 5.5: The MRE of backpropagation with momentum and adaptive learning rate ANN using all the data. In bold are the lowest MRE.

Epochs	Vmin	Vmax	H	Carr	DT	Average
10	8.995	6.850	3.699	7.546	14.102	8.239
20	8.146	6.678	3.840	7.838	13.543	8.009
40	8.118	3.924	4.034	6.668	11.952	6.939
80	4.369	3.979	2.689	5.080	12.830	5.790
160	3.566	3.967	2.782	5.826	9.114	5.051
320	2.451	3.249	**2.464**	**4.842**	11.017	4.805
640	**2.245**	**2.910**	2.478	4.951	9.165	4.350
1280	2.538	3.015	2.481	4.874	**7.213**	**4.024**
2560	2.482	2.966	2.607	5.112	7.512	4.136

It is shown in Tables 5.4 & 5.5 that there is no ability to predict the validation results by training ANN of each topology tested on all the data. This conclusion stems from the fact that there is no correlation between the results in Table 5.4 and the corresponding results in Table 5.5 (i.e. the best ANN models of Table 5.4 and Table 5.5 are different). Hence, it is not possible to choose the number of epochs according to this method. In this study as opposed to the previous one, reducing complexity of the network, by not including the other responses, contributed to the network performance. The difference between the two studies is probably due to the relationships between the responses modelled. It is probable that disintegration time is less related to the other responses in the capsule data whereas the relationship between dissolution rate (which was modelled separately) and the other responses of the tablet study is stronger.

It is obvious there was a lack of data hence the ability to predict only 5 out 9 response variables. This lack of data is due to big 'holes' in the hypersurface. A second reason is related to the connection between the type of response to the independent variables. As an example, it might be that variance of dissolution time cannot be predicted by this set of independent variables regardless of the amount of data used for modelling. It might be that variables that are important for modelling responses, like variance of dissolution time as an example, are not included since they are uncontrolled or unnoticed. It is not surprising that it was possible to predict minimum bulk density, maximum bulk density as well as Carr's compressibility index and Hausner's ratio, since the latter two are calculated from the former ones (see Background chapter).

Hogan et al., (1996) succeeded to model the data using nonparametric canonical correlation analysis not only for the responses that the present study succeeded to model but also to the responses of coefficient of fill weight variation and variance of dissolution time. Hogan used a technique that is primarily descriptive although it may be used for predictive purposes. The multiple coefficient of determination in regression (R^2) is equivalent to interranging communalities (d^2) (Douglas & Love, 1968). They both measure the amount of variance in the dependent variable explained by the model. But it cannot be said like in regression that if $d^2 = 1$, then 100% of the variance in the dependent variable set is available to be explained by the independent variable set. This is since unlike in multiple regression it does not deal with a single dependent variable. Hence, it is not possible to compare the R^2 values (see Appendix B) of this study with the d^2 values of Hogan et al. (1996). Appropriate comparison could be made by comparing the predictive ability of the two methods using the same validation method, as was used in the comparison of ANN versus regression in this study.

Table 5.6: The prediction's relative errors (in percent) of regression and ANN. Each value was predicted by excluding it from the model using the leave-one-out method.

Case	A_Vmin	A_Vmax	A_H	A_Carr	A_DT	R_Vmin	R_Vmax	R_H	R_Carr	R_DT
1	-6.54	4.87	6.71	16.57	-4.69	-8.47	3.10	7.45	13.22	8.41
2	0.21	5.41	4.89	8.19	1.97	-1.31	2.83	4.38	8.06	-10.26
3	-2.79	0.33	4.06	6.40	5.77	-2.44	3.11	4.38	8.33	1.66
4	-10.97	-3.60	5.78	12.55	-3.32	-9.02	3.73	4.38	8.20	1.37
5	-1.55	2.04	1.22	4.34	-12.95	-1.30	3.32	0.44	1.03	-4.68
6	2.67	-3.29	-5.58	-9.89	-8.07	1.02	-2.26	-5.33	-11.30	-11.29
7	-2.95	-3.89	-2.14	-5.08	-0.84	0.09	-3.75	-2.36	-4.77	-15.74
8	0.01	-3.94	-1.32	-7.21	1.31	-0.62	-5.28	-1.65	-3.21	0.65
9	1.33	-2.25	-3.44	-7.46	-27.08	0.60	-2.26	-3.09	-6.30	-11.49
10	4.21	1.33	-1.70	-0.30	1.62	2.69	0.61	-1.65	-2.40	-14.23
11	1.50	-3.43	-6.02	-13.12	-0.92	2.69	-3.75	-6.09	-12.36	-6.52
12	-3.42	-1.15	1.48	0.61	2.14	-1.98	0.61	1.79	3.08	4.58
13	-3.10	-0.82	-0.32	2.33	-18.58	0.15	0.26	0.27	1.69	-19.12
14	-1.44	1.08	-2.30	-0.89	6.04	6.03	4.20	-1.73	-2.58	3.30
15	1.28	1.33	-0.54	-1.03	22.04	0.50	0.61	-0.25	-0.28	12.35
16	13.01	4.75	-2.10	-3.87	-1.87	-8.14	-6.12	2.13	0.92	23.27
17	-3.98	2.57	-0.52	1.19	-7.34	22.91	8.34	-0.25	-0.37	-48.41
18	3.11	0.75	-1.88	-6.86	-3.23	-4.44	-8.55	-6.87	-11.81	6.57
19	0.79	-2.13	-1.07	-2.58	-2.97	-1.26	-2.54	-3.09	-3.16	-4.09
20	-4.55	-4.01	-1.64	-5.93	-25.44	-2.55	-4.26	-0.25	-5.23	-32.42
21	-2.22	-5.00	0.99	-7.40	29.64	-5.21	-9.62	1.12	-8.38	18.33
22	4.91	4.62	-0.58	-2.93	27.68	4.06	3.05	-1.06	-1.96	16.94
23	2.15	4.43	2.67	6.17	-4.82	1.12	3.85	3.78	7.29	-13.02
24	-3.04	-13.77	-8.43	-16.12	-9.63	-6.44	-14.07	-7.22	-15.15	-14.83
25	-4.64	3.97	5.35	12.37	-2.00	-3.56	6.41	7.18	12.18	1.69
26	-4.48	2.89	6.79	10.09	-8.96	-5.34	-0.80	5.00	8.44	-13.77
27	2.16	0.89	1.38	0.98	8.55	3.09	-2.26	-1.85	-3.82	6.45
28	-1.95	-1.81	1.76	2.01	15.16	-2.74	-8.49	-2.08	-2.69	11.41
29	2.38	6.13	2.60	0.11	-25.02	3.86	-0.80	-0.49	-1.16	-15.32
30	6.57	7.61	2.05	0.60	19.12	4.89	6.88	1.87	4.05	20.18
31	1.34	0.51	2.39	4.89	-6.13	2.47	3.05	1.87	3.49	13.12
32	2.57	-2.98	1.62	-4.30	24.93	4.25	0.09	-2.78	-3.82	23.50
33	13.88	9.53	-3.07	-9.04	-6.32	12.16	8.39	-0.14	-0.36	21.24

Note. The abbreviations for the responses stand for (from second left to right column): minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time respectively. This set is repeated twice for ANN ("A_") and regression ("R_") respectively.

The following table summarises the MRE results. The abbreviations in Table 5.7 relate to the following responses respectively (form second left column to right one): minimum bulk density (Vmin), maximum bulk density (Vmax), Hausner's ratio (H), Carr's compressibility index (Carr) and disintegration time (DT).

Table 5.7: Average percentage deviation between predicted and observed values for the best models of regression analysis and ANN.

Model	Vmin	Vmax	H	Carr	DT
Reg.	4.16	4.16	2.86	5.49	13.04
ANN	3.69	3.55	2.86	5.86	10.49

Note. The abbreviations in the table above relate to the following responses: minimum bulk density (Vmin), maximum bulk density (Vmax), Hausner's ratio (H), Carr's compressibility index (Carr) and disintegration time (DT).

5.3.1.1 Optimising the number of epochs and hidden neurons

In order to optimise ANN performance the number of epochs was varied as well as the number of hidden neurons to get the best ANN model, as was mentioned before in section 5.2. The ANN MRE results presented previously on Table 5.4 summarise the results of the backpropagation with momentum and adaptive learning rate training method. It does not however present all on that particular training method since it does not present the results for each topology but only the best ones. All the results, except the ones that relate to 20000 iterations, for the response of disintegration time are presented in Figure 5.2.

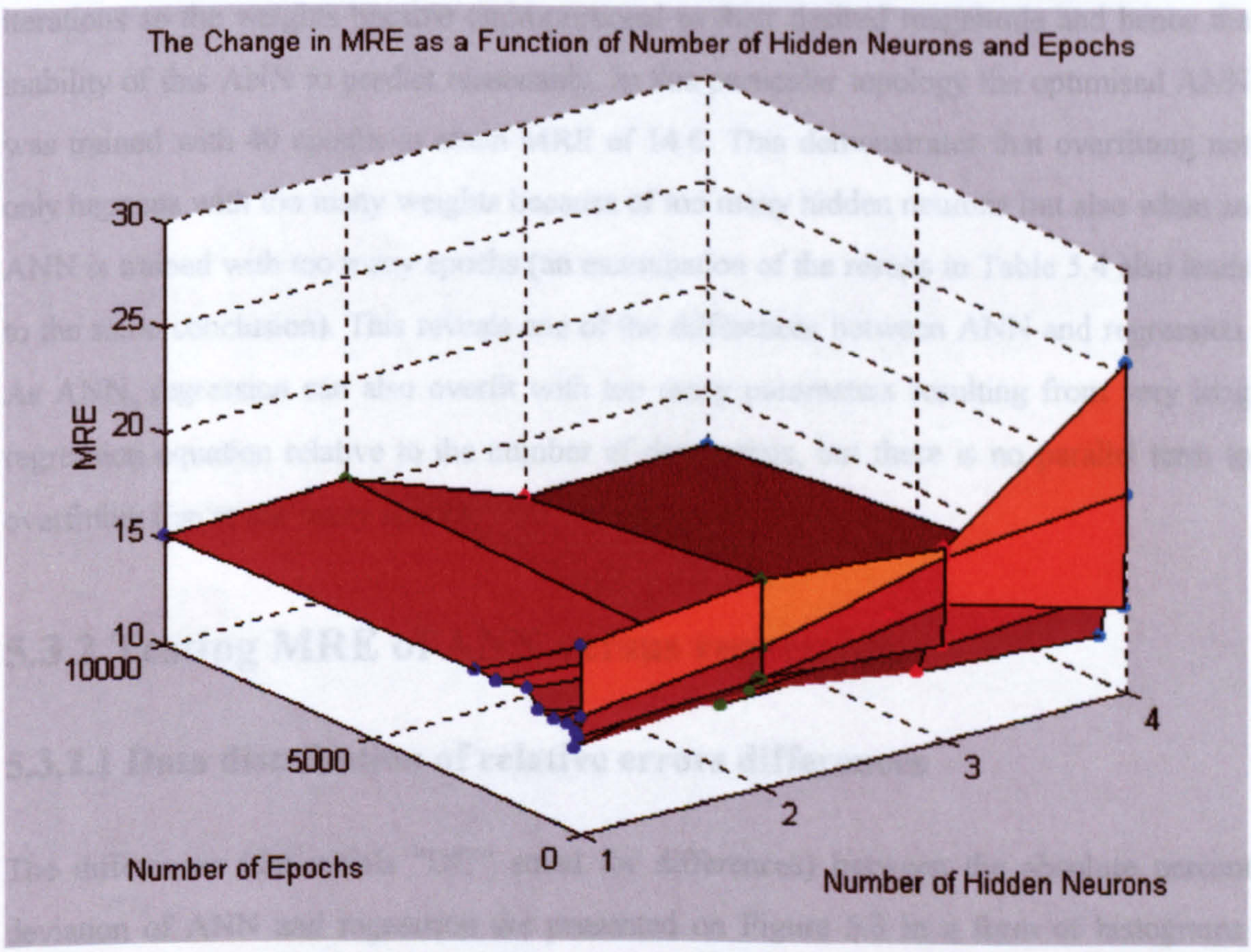


Figure 5.2: The change in MRE as a function of number of hidden neurons and epochs for the disintegration time response.

Looking at Figure 5.2, the coloured circles are the actual values whereas the surfaces were generated using interpolation between these circles. On the bottom right corner one can see the minimum point with ANN set-up of 4 hidden neurons and 640 epochs to achieve MRE of 12.144. As mentioned before the best ANN for this response employed just one output neuron (the plot above considered just the disintegration predictions from ANN of several output neurons) with MRE of 10.488.

Each point in Figure 5.2 represents an average of percent deviations from predictions generated by running the ANN 33 times and excluding one each time. When ANN was trained for 20,000 epochs it generated reasonable results for topology 2, 3, 4 with MRE of 15.522, 21.584 and 15.615 respectively. The MRE of the topology with only one hidden neuron was 2.31×10^{114}. This high number was due to only one trained ANN out of all the 33, so that when one case was excluded, trying to predict its value generated percent relative error of that high order. This one specific ANN modelled several responses simultaneously but was given just one hidden neuron so it did not have enough weights to model the problem. In addition for this small number of weights the ANN performs many iterations so the weights became unproportional to their desired magnitude and hence the inability of this ANN to predict reasonably. In this particular topology the optimised ANN was trained with 40 epochs to attain MRE of 14.6. This demonstrates that overfitting not only happens with too many weights because of too many hidden neurons but also when an ANN is trained with too many epochs (an examination of the results in Table 5.4 also leads to the same conclusion). This reveals one of the differences between ANN and regression. As ANN, regression can also overfit with too many parameters resulting from very long regression equation relative to the number of data points, but there is no parallel term to overfitting due to too many epochs.

5.3.2 Testing MRE of ANN versus regression

5.3.2.1 Data distribution of relative errors differences

The differences (the initials "DIF" stand for differences) between the absolute percent deviation of ANN and regression are presented on Figure 5.3 in a form of histograms. There is normal distribution line on these graphs in order to inspect how the data distribution resembles normal distribution. The following responses are presented respectively, minimum & maximum bulk density, Hausner's ratio, Carr's compressiblity index and disintegration time. For more robust examination of data distribution, Shapiro-Wilk test that is a quantitative statistical test for normality is presented in Table 5.8.

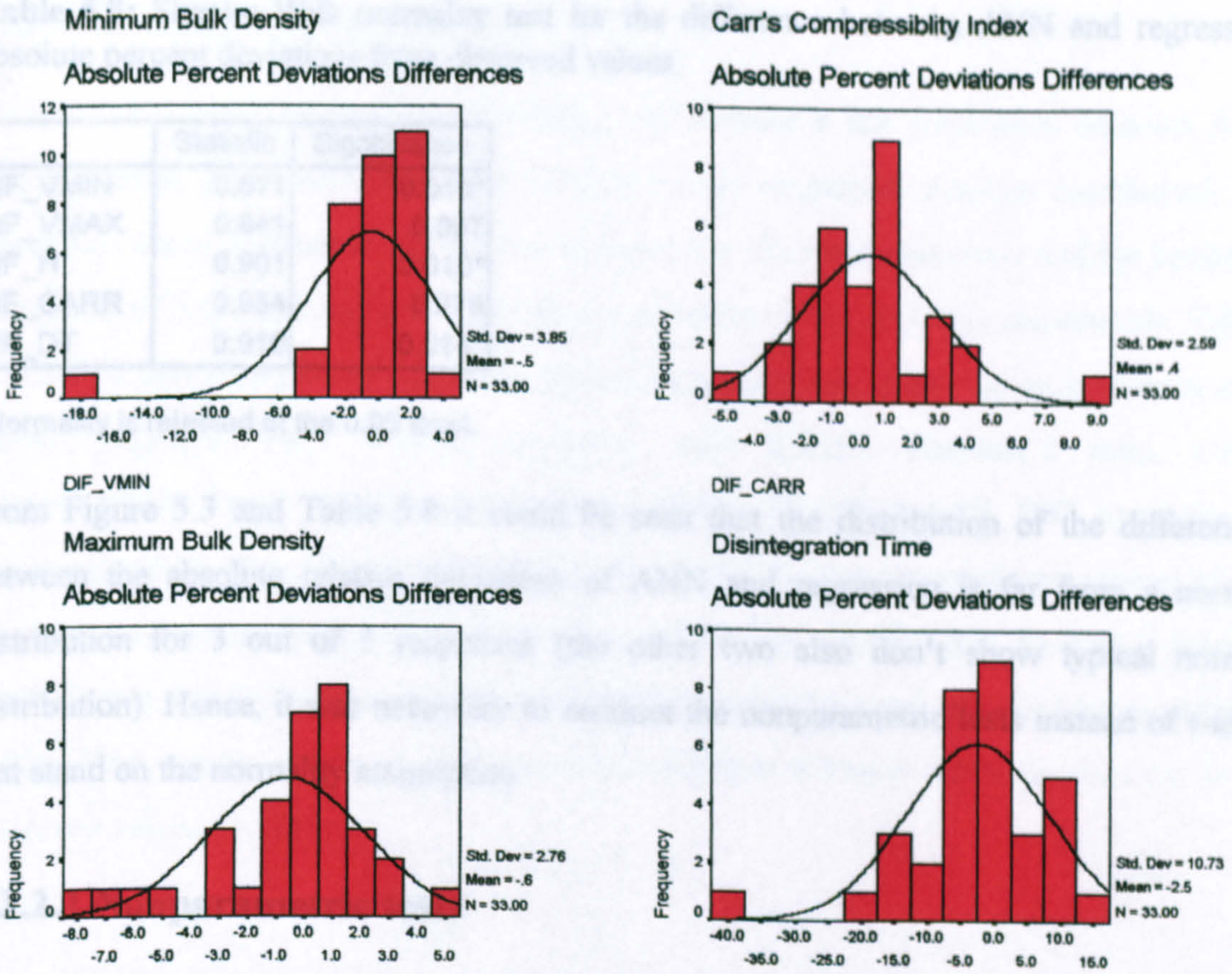


Figure 5.3: Histograms for the differences between regression and ANN absolute percent deviations from observed values. The bars represent the actual values whereas the black line displays the normal distribution.

Note. The plots are for the following responses respectively (from top figure to bottom one and from left to right column of figures): minimum & maximum bulk density, Hausner's ratio, Carr's compressiblity index and disintegration time.

Table 5.8: Shapiro-Wilk normality test for the difference between ANN and regression absolute percent deviations from observed values.

	Statistic	Significance
DIF_VMIN	0.671	0.010*
DIF_VMAX	0.941	0.097
DIF_H	0.901	0.010*
DIF_CARR	0.954	0.278
DIF_DT	0.910	0.014*

*Normality is rejected at the 0.05 level.

From Figure 5.3 and Table 5.8 it could be seen that the distribution of the differences between the absolute relative deviations of ANN and regression is far from a normal distribution for 3 out of 5 responses (the other two also don't show typical normal distribution). Hence, it was necessary to conduct the nonparametric tests instead of t-tests that stand on the normality assumption.

5.3.2.2 Nonparametric tests

To test whether ANN predicts better than regression, Wilcoxon signed ranks tests were performed on absolute percent deviation (MRE) of ANN versus regression, and are presented in Table 5.9.

Table 5.9: Nonparametric tests, of Wilcoxon signed ranks test, to test whether ANN predicts better than regression. "Dif" stand for the difference between ANN and regression (ANN minus regression).

Response	Mean_Dif	Median_Dif	Significance
V_MIN	-0.476	0.250	0.521
V_MAX	-0.610	-0.130	0.235
H	2.860	2.100	0.643
CARR	0.373	0.480	0.695
DT	-2.547	-1.120	0.136

According to the results presented in Table 5.9 the nonparametric tests did not show ANN predicts better than regression. The most significant advantage ANN has is in its predictive ability of disintegration time response (but still, it is not statistically significant).

5.3.3 Bivariate correlation

To determine which method predicts better, and if there is any corrleation between ANN and regression predictions, bivariate correlation was employed. Pearson correlations and Spearman's non-parametric correlations between the observed responses and the predicted responses of ANN and regression are shown in Tables 5.10a & 5.10b respectively. Tables 5.10a & 5.10b relate to the following responses respectively (from top to bottom in each table): minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time. "DT", "A_DT" and "R_DT" stand for the observed values, the ANN predicted values and the regression-predicted values of the disintegration time response respectively. To read the data, for example, looking at the first table (for minimum bulk density response) on the second row and column gives a value of 0.889. That means, the Pearson correlation coefficient between ANN predictions to the observed values is 0.889.

Table 5.10a: Pearson correlations between the responses and the predicted responses of ANN and regression.

	V_MIN	A_V_MIN	R_V_MIN
V_MIN	1.000	.889**	.885**
A_V_MIN	.889**	1,000	.767**
R_V_MIN	.885**	.767**	1.000
	V_MAX	A_V_MAX	R_V_MAX
V_MAX	1.000	.877**	.915**
A_V_MAX	.877**	1.000	.690**
R_V_MAX	.915**	.690*	1.000
	H	A_H	R_H
H	1.000	.635**	.935**
A_H	.635**	1.000	.393*
R_H	.935**	0.393*	1.000
	CARR	A_CARR	R_CARR
CARR	1.000	.590**	0.933**
A_CARR	.590**	1.000	0.296
R_CARR	.933**	0.296	1.000
	DT	A_DT	R_DT
DT	1.000	.518**	0.779**
A_DT	.518**	1.000	0.046
R_DT	.779**	0.046	1.000

*. Correlation is significant at the .05 level (1-tailed).

**. Correlation is significant at the .01 level (1-tailed).

Note. The tables above relate to the following responses respectively: minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time respectively, e.g. "DT", "A_DT" and "R_DT" stand for the observed values, the predicted values by ANN and the predicted values by regression regarding the disintegration time response respectively.

Table 5.10b: Spearman's non-parametric correlations between the responses and the predicted responses of ANN and regression.

	V_MIN	A_V_MIN	R_V_MIN
V_MIN	1.000	.931**	.934**
A_V_MIN	.931**	1.000	.807**
R_V_MIN	.934**	.807**	1.000
	V_MAX	A_V_MAX	R_V_MAX
V_MAX	1.000	.881**	.917**
A_V_MAX	.881**	1.000	.738**
R_V_MAX	.917**	.738**	1.000
	H	A_H	R_H
H	1.000	.510**	.964**
A_H	.510**	1.000	.384*
R_H	.964**	.384*	1.000
	CARR	A_CARR	R_CARR
CARR	1.000	.452**	.962**
A_CARR	.452**	1.000	.275
R_CARR	.962**	.275	1.000
	DT	A_DT	R_DT
DT	1.000	.410**	.696**
A_DT	.410**	1.000	-.046
R_DT	.696**	-.046	1.000

*. Correlation is significant at the .05 level (1-tailed).

**. Correlation is significant at the .01 level (1-tailed).

Note. The tables above relate to the following responses respectively: minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time respectively, e.g. "DT", "A_DT" and "R_DT" stand for the observed values, the predicted values by ANN and the predicted values by regression regarding the disintegration time response respectively.

A quick view of all the results in Table 5.10 (parametric and non-parametric) reveals higher correlation coefficients for regression relative to ANN. However, both ANN & regression models were significant in all the tests of correlation coefficients at the 99% confidence level. The correlation coefficients (both parametric and non-parametric) for the relation between ANN and regression have confidence level of more than 99% for the responses of minimum and maximum bulk density, 95% for Hausner's ratio and less than 95% for Carr's index (weak correlation). For the disintegration time, the correlation coefficients show values that demonstrate there is no correlation between ANN and regression predictions.

5.3.4 Precision test of ANN versus regression

To compare the precision of ANN versus regression, F-Tests for variances of percent relative errors (signed MRE) between ANN and regression were conducted, and the results are presented in Table 5.11. In the left column is the name of the statistical measurement/test involved. The other columns present the values of these measurements/tests. The abbreviations for the responses stand for (from left to right column and continued below): minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time respectively. "A_" and "R_" stand for initials that relate to ANN and regression percent relative deviation respectively.

Table 5.11: F-Test for variances of percent relative errors between ANN and regression.

	A_V_min	R_V_min	A_V_max	R_V_max	A_H	R_H
Mean	0.20	0.23	0.39	-0.37	0.28	-0.07
Variance	24.80	37.25	20.61	28.07	12.95	13.60
F	0.67		0.73		0.95	

	A_Carr	R_Carr	A_DT	R_DT
Mean	-0.44	-0.64	-0.43	-1.22
Variance	56.68	48.83	202.18	270.21
F	1.16		0.75	

Note. The abbreviations for the responses stand for (from left to right column and from top table to bottom one): minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time respectively. "A_" and "R_" stand for initials that relate to ANN and regression percent relative deviation respectively.

Looking at the precision test results on Table 5.11, it can be seen there are no major differences between the precision of the two methods. The biggest difference is in the response of minimum bulk density. In this response, the variance of ANN percentage relative errors is 24.80, the variance of regression percentage relative errors is 37.25 and the F-value is 0.67. Hence, ANN is more precise than regression by about one third ("about" because F-test is done on independent samples and not paired data as in this study, so it is not possible to use significance values generated in this test).

5.3.5 Bias examination

To examine if the predicted values generated by ANN and regression models are significantly below or above the observed values, bias tests were conducted. Bias tests on signed residuals (observed minus predicted) of ANN ("A_") and regression ("R_") are shown in Table 5.12. The left-hand column gives information on whether it is ANN or regression modelling type and the response modelled. The abbreviations for the responses stand for (from the top): Carr's compressibility index, disintegration rate, Hausner's ratio, maximum and minimum bulk density respectively. This set of abbreviations is repeated since it relates to ANN as well as regression.

Table 5.12: Bias test on signed residuals (observed minus predicted) of ANN ("A_") and regression ("R_").

One-Sample Test

	Test Value = 0					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
A_CARR	,009	32	,992	4,364E-03	-,9323	,9411
A_DT	,397	32	,694	9,315E-02	-,3854	,5717
A_H	,638	32	,528	6,136E-03	-1,35E-02	2,574E-02
A_V_MAX	,821	32	,418	4,456E-03	-6,60E-03	1,551E-02
A_V_MIN	,439	32	,664	1,885E-03	-6,86E-03	1,063E-02
R_CARR	-,188	32	,852	-7,92E-02	-,936947	,77852518
R_DT	-,136	32	,892	-3,73E-02	-,595663	,52096649
R_H	,072	32	,943	,00070838	-,019368	,02078484
R_V_MAX	-,229	32	,820	-1,47E-03	-1,5E-02	1,16E-02
R_V_MIN	,308	32	,760	1,64E-03	-9,2E-03	1,25E-02

Note. The abbreviations from top downwards stand for responses of: Carr's compressibility index, disintegration time, Hausner's ratio, maximum and minimum bulk density respectively. This set of abbreviations is repeated twice since it relates to ANN as well as regression.

The bias results presented on Table 5.12 reveals there is no model with a significance value which is so low that the null hypothesis (according to the null hypothesis there is no bias) could be rejected. Hence, there is no bias in any of the models. One could also examine the 95% confidence interval and see that the 0 value lies within the interval of all the models.

5.3.6 The predictive ability as a function of the response value

As was mentioned in the previous chapter, sometimes one model is better on a certain response domain than the other. What follows is a presentation of two sets of graphs to detect the relationship between ANN and regression errors in predictions to the response values.

Figure 5.4 shows the difference between the absolute residuals (observed-predicted) of regression minus those of ANN on the y-axis, as a function of the observed value on the x-axis. The initials "RD" on the y-axis stands for the residual difference. The plots of Figure 5.4 relate to the following responses, from top to bottom, respectively: minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time. Figure 5.5 is the same plot as the one that relates to disintegration time in Figure 5.4 with one exception - one outlier data point is omitted. Figure 5.6 demonstrates the dependency of the percent relative error (MRE) on the y-axis upon the response value on the x-axis. The MRE data was taken from Table 5.6, omitting the sign to get the absolute values. The red squares are ANN relative errors whereas the red line is the least-square regression line which demonstrates trend, similarly green relate to regression data points and trend line.

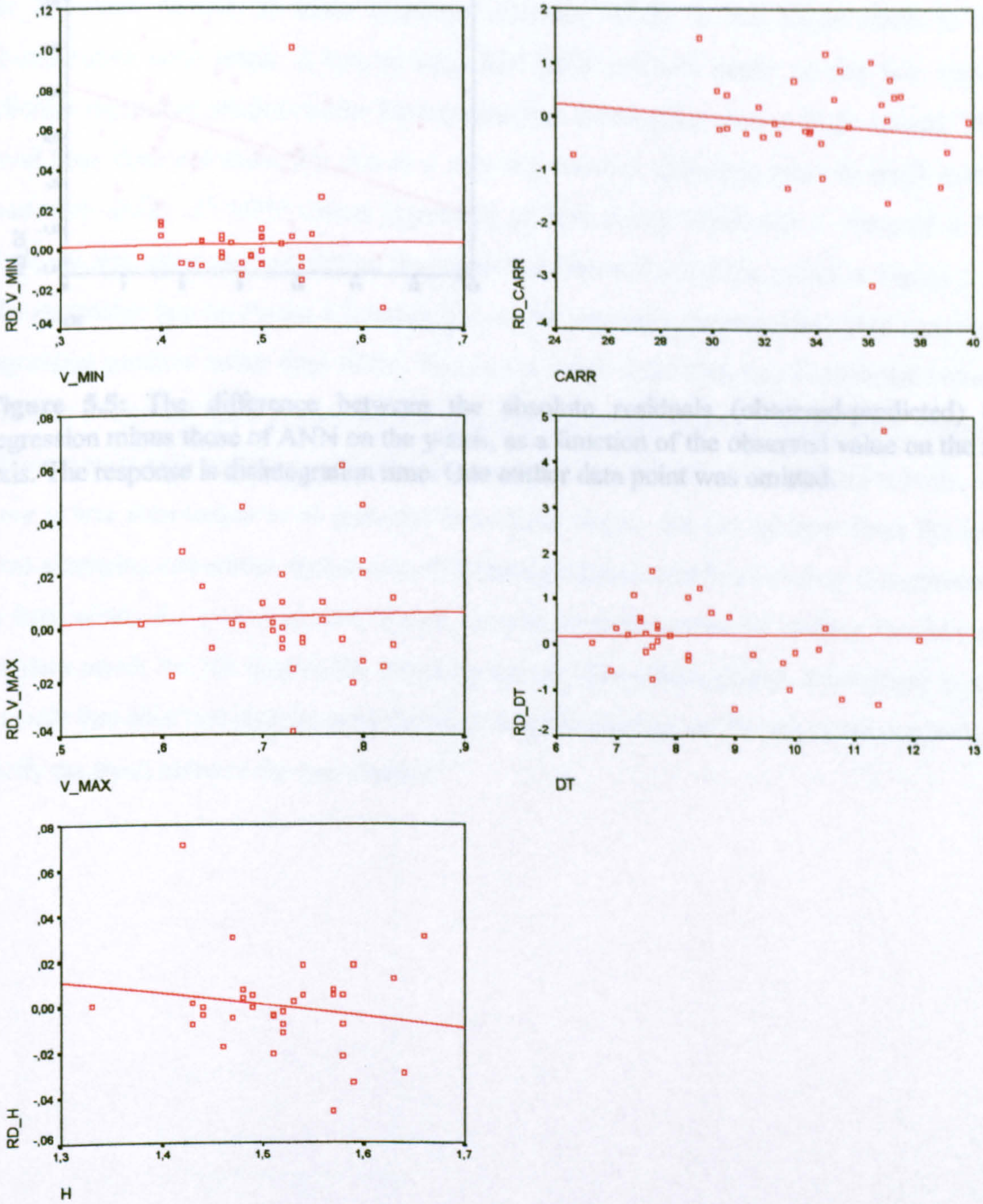


Figure 5.4: The difference between the absolute residuals (observed-predicted) of regression minus those of ANN on the y-axis, as a function of the observed value on the x-axis.

Note. The figures above relate to the following responses respectively (from top figure to bottom one and from left to right column of figures): minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time.

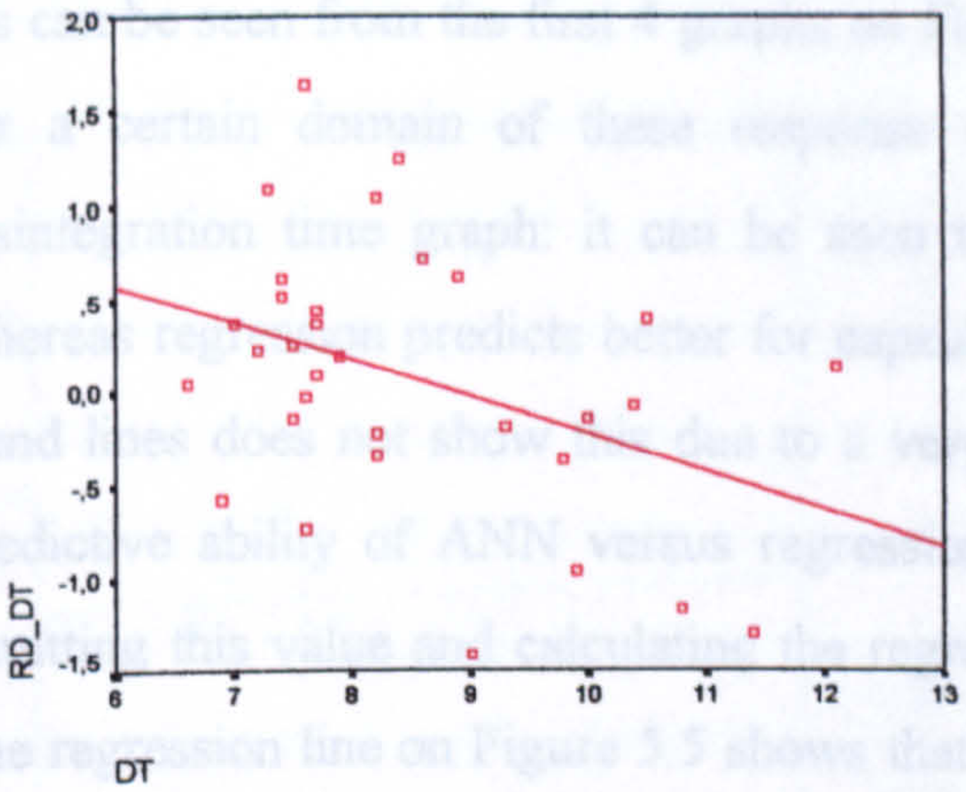


Figure 5.5: The difference between the absolute residuals (observed-predicted) of regression minus those of ANN on the y-axis, as a function of the observed value on the x-axis. The response is disintegration time. One outlier data point was omitted.

As can be seen from the first 4 graphs on Figure 5.4, there is no one method that is better for a certain domain of these response variables. There is one slight trend at the disintegration time graph: it can be seen that ANN predicts better on the low values whereas regression predicts better for capsules that disintegrate slower (high values). The trend lines does not show this due to a very big residual difference (due to much better predictive ability of ANN versus regression on this point) which has a value of 4.72. Omitting this value and calculating the regression line will yield the graph in Figure 5.5. The regression line on Figure 5.5 shows that as the capsule's disintegration time increases regression predicts better than ANN. This is the same trend that was mentioned before. Hausner's ratio and Carr's index show negative slope mainly due to 2 extreme big residual differences in each response with the values of -0.05, 0.07 and -3.1, 1.5 respectively, so there is less information to be gathered from these slopes. As can be seen from the last three examples, sometimes outliers can shift the regression trend line in a way that prevents us from seeing the overall picture. Hence, there is no replacement for looking carefully at the data points on the graphs for detecting trends. The disintegration time graph is an example that once two clusters were detected, a regression line can be only used as a tool to clarify the trend between the two clusters.

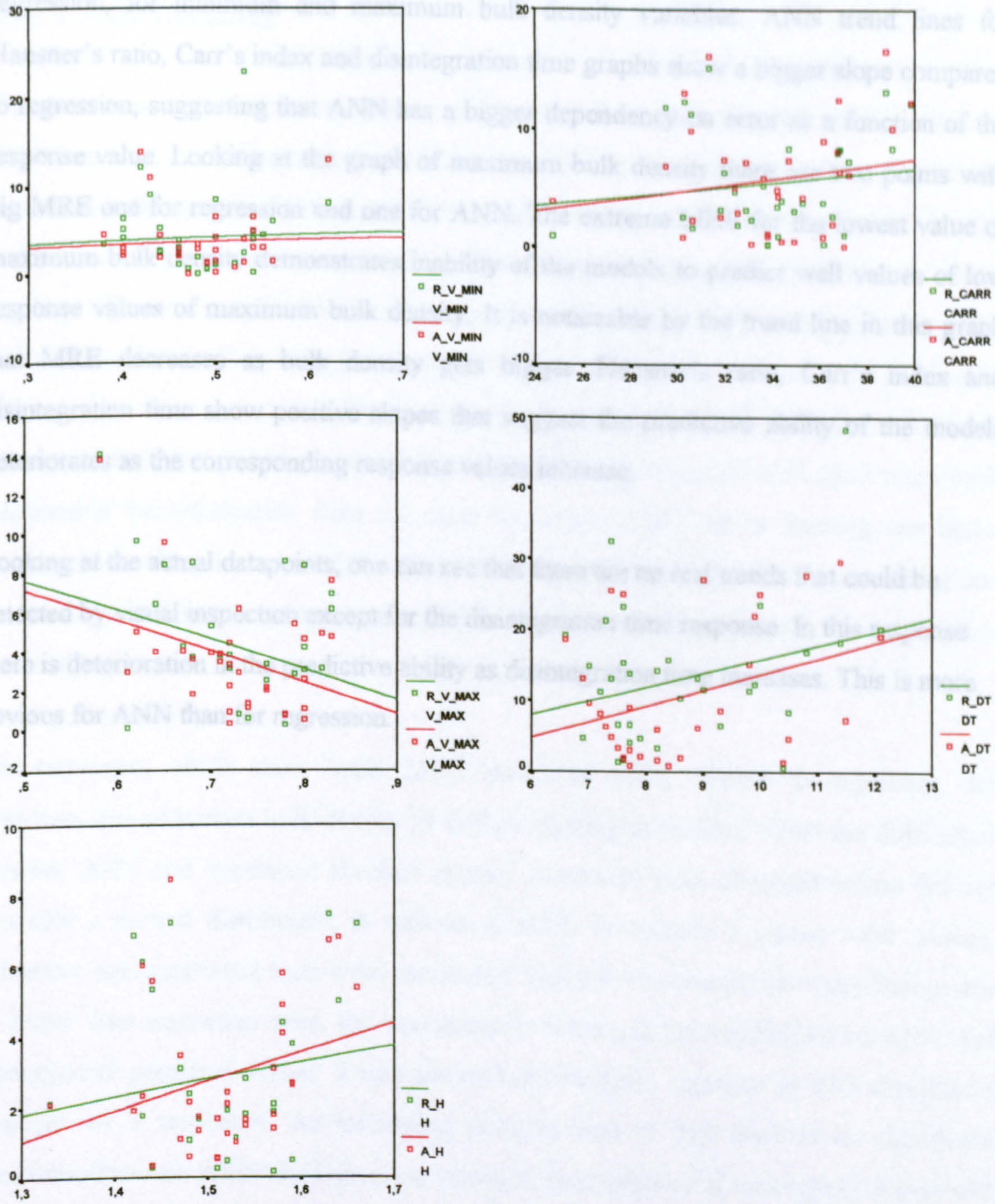


Figure 5.6: Percent relative error (in absolute values) as a function of the response value. In red and green are ANN and regression results respectively. The regression lines represent the trend.

Note. The figures above relate to the following responses respectively (from top figure to bottom one and from left to right column of figures): minimum bulk density, maximum bulk density, Hausner's ratio, Carr's compressibility index and disintegration time.

Examination of the graphs in Figure 5.6 show almost the same slope for ANN and regression, for minimum and maximum bulk density variables. ANN trend lines for Hausner's ratio, Carr's index and disintegration time graphs show a bigger slope compared to regression, suggesting that ANN has a bigger dependency on error as a function of the response value. Looking at the graph of maximum bulk density there are two points with big MRE one for regression and one for ANN. The extreme MRE for the lowest value of maximum bulk density demonstrates inability of the models to predict well values of low response values of maximum bulk density. It is noticeable by the trend line in this graph that MRE decreases as bulk density gets bigger. Hausner's ratio, Carr's index and disintegration time show positive slopes that suggest the predictive ability of the models deteriorates as the corresponding response values increase.

Looking at the actual datapoints, one can see that there are no real trends that could be detected by visual inspection except for the disintegration time response. In this response there is deterioration in the predictive ability as disintegration time increases. This is more obvious for ANN than for regression.

5.4 Conclusions

There is no possibility of saving time by predicting MRE results through running ANN on all data points as can be seen from looking at Tables 5.4 and 5.5. In the response of minimum bulk density ANN is about one third more precise than regression. On the response of disintegration time a topology of one output neuron gave better results than a topology of several output neurons. By using just one output neuron the complexity of the ANN declined since there were fewer weights and they did not have to take into account all responses, in this respect it is like the regression model. Probably there is not a strong correlation between disintegration time to the other responses, since if there were such relationship the information from the other responses would aid in learning and better predictions resulting in lower MRE of the more complex topology. Hence, not taking into account all ANN responses can cause ANN to learn better depending on the type of data involved.

The responses, which show better MRE results of ANN relative to regression, are minimum and maximum bulk density as well as disintegration time. Since the differences between ANN and regression absolute percent deviations from observed values did not resemble a normal distribution, it was not possible to conduct a paired t-test. Hence, Wilcoxon non-parametric tests were conducted. These tests showed ANN predictions are no better than regression ones. On examinations of correlation coefficients of ANN and regression to predicted values, it was shown both methods correlate in 99% confidence level for all 5 responses. An interesting point to note is that there is no significant correlation between ANN and regression values in the responses of disintegration time and Carr's compressiblity index. That means, that regarding the latter responses using an ANN will produce quite different predictions than using regression.

Regarding the relation between the response values and the errors in predictions for both ANN and regression two phenomena were detected. Regarding disintegration time response, ANN and regression relate to this response in such a way that ANN predicts better for faster disintegrating capsules whereas regression predicts better for the slower disintegrating ones. ANN is more susceptible to this response value than regression since it shows greater deterioration in the predictive ability as the response value increases.

The regression models tested here were very limited since they did not include interaction and squared terms as in the tablet study. So there may not have been enough flexibility to model the system. This may explain the difference in the predictive ability of ANN versus regression for the disintegration time response, although this was not found to be statistically significant. To verify this conclusion, a more complicated regression model was constructed. Its starting equation included all squared terms and also interaction terms between filler type and filler level, disintegrant type and disintegrant level, lubricant level and glidant level, and all possible interactions concerning drug properties. Backward-elimination that was done on this equation yielded an equation that could generalise better than the best ANN model, since using the leave-one-out for validation on this equation yielded MRE value of 8.66!

6. Multiobjective Optimisation of Tablet Formulations and Processes

6.1 Introduction

Achieving the desired properties for a tablet is important for a number of reasons. This is an imperative primarily for regulatory approval since in order to get approval for the product it must have adequate properties, e.g. disintegrate fast enough. Another reason may be that the regulatory requirements are made more stringent and the product needs improvement. The advantage of tablet properties that exceed the requirements of the regulatory authorities lies in increasing the chance that more time elapses before there is a need for improvement, if any improvement is required at all. Doing optimisation and achieving goals that are beyond what is needed, alleviate the quality of the product and it is good both for the consumer and for the company, e.g. increase in dissolution rate could cause the user to have a relief from pain (in case of pain-killers) faster. Another issue is that optimisation of tablet properties saves time and money by decreasing the number of experiments. There is also another advantage of being able to develop the formulation faster and maybe to be first on the market. This is an important marketing principle that could yield money for a pharmaceutical company.

From the nature of pharmaceutical formulation it is clear that problems need to be addressed in a manner that takes into account all the relevant responses. For example it is

no use optimising formulation and process parameters to improve dissolution rate response without taking into account tablet hardness. Increasing the compaction force will usually cause the tablet to be harder and dissolve slowly and vice versa. The essence of the pharmaceutical formulation and process problem is hence the optimisation of conflicting variables, and to resolve this multiobjective optimisation is required.

There are number of multiobjective optimisation techniques used for optimisation of pharmaceutical formulations. Graphical techniques were commonly used for optimisation in the pharmaceutical field (Schwartz et al., 1973). These have evolved to numerical methods like the Monte Carlo approach, which uses random number techniques to reach the optimum solution (Takayama & Nagai, 1989). Building a regression equation and optimising using graphical methods and/or simplex algorithm (Turkoglu et al, 1992; see Edwin & Stanislaw, 1996 for a description of the simplex algorithm and other optimisation algorithms) is one common approach that symbolises the more general approach of building a model and optimisation. There is also a method that does not use any model, this is dynamic multiobjective optimisation (Shek et al, 1980). The dynamic optimisation is an iterative process of feeding the computer the latest experiment and it suggests a new experiment that is supposedly closer to the optimum solution point. With the emergence of the use of ANN in pharmaceutical formulation development these ANN too were incorporated into the process of optimisation. There is the direct approach of inversion of a trained ANN (Turkoglu et al., 1995; Achanta et al., 1995) but the more popular methods involve a combination of ANN and optimisation methods e.g. a combination of ANN and a genetic algorithm (Dowell et al., 1997/1999). The latter approach was implemented in Cad/Chem software for formulation optimisation (Colbourn & Rowe, 1996). Another approach is model building using regression/ANN and multiobjective optimisation by a generalised distance function method. This was done in several studies relating to solid dosage forms (Takahara et al., 1997a) as well as the formulation of gels (Takahara et al., 1997b). The last two articles by Takahara and co-workers used an ANN and an optimisation algorithm whereas a combination of regression with multiobjective optimisation was the methodology in earlier papers (Levison et al., 1994, used this approach to optimise formulation of gels). There is also another variation of the multiobjective optimisation method that enables one to give preferences for the different responses to be optimised (Takayama & Nagai, 1991). This study will present a novel method for multiobjective optimisation that incorporates ANN for building the model and

the Goal Attainment method of Gembicki (1974) for optimisation. The Goal Attainment method can also take into account the priorities of the responses to be optimised. The next sections explain this method.

The aim of this study was to minimise disintegration time and impact friability simultaneously by creating and using a novel computer program that implements a new optimisation algorithm. The combination of ANN with the Goal Attainment method (Gembicki, 1974) for optimisation was the new algorithm implemented. The responses were arbitrarily chosen with the guideline being that their optimisation goals are conflicting. Only two responses were chosen to simplify this study and to avoid too much detail, but all 8 responses could be optimised simultaneously in the same manner. An example of this multiobjective optimisation is set out below.

In this example the object is to minimise $F(\mathbf{x})$. $\mathbf{x}$ consists of x_1, x_2 and x_3 representing the independent parameters of lubricant (%), disintegrant level (%) and compaction force (kN) respectively. The feasible region is defined as Ω. $\mathbf{x}$ belongs to the three dimensional parameter space - $\Re^3$, briefly denoted as $\mathbf{x} \in \Re^3$, So $\Omega = \{ \mathbf{x} \in \Re^3 \}$.

The constraints on $\mathbf{x}$ were:

$0.1 \leq x_1 \leq 100$
$1.0 \leq x_2 \leq 100$
$3.0 \leq x_3 \leq 100$

The corresponding feasible region is generated using the trained ANN. The trained ANN is the function F which generates the appropriate solution according to $\mathbf{x}$. For each $\mathbf{x}$ value there is an appropriate $\mathbf{y} = F(\mathbf{x})$ value in Figure 6.1 which defines the feasible region for the objective function, this space is denoted by Λ. Since there are two responses to be optimised the objective function is in two dimensional parameter space. It can be summarized as:

$\Lambda = \{ \mathbf{y} \in \Re^2 \}$ where $\mathbf{y} = F(\mathbf{x})$ subject to $\mathbf{x} \in \Omega$.

Note that Figure 6.1 does not represent the true $F(\mathbf{x})$ values for the optimisation problem presented in this work, rather it is a depiction of an arbitrary objective function region.

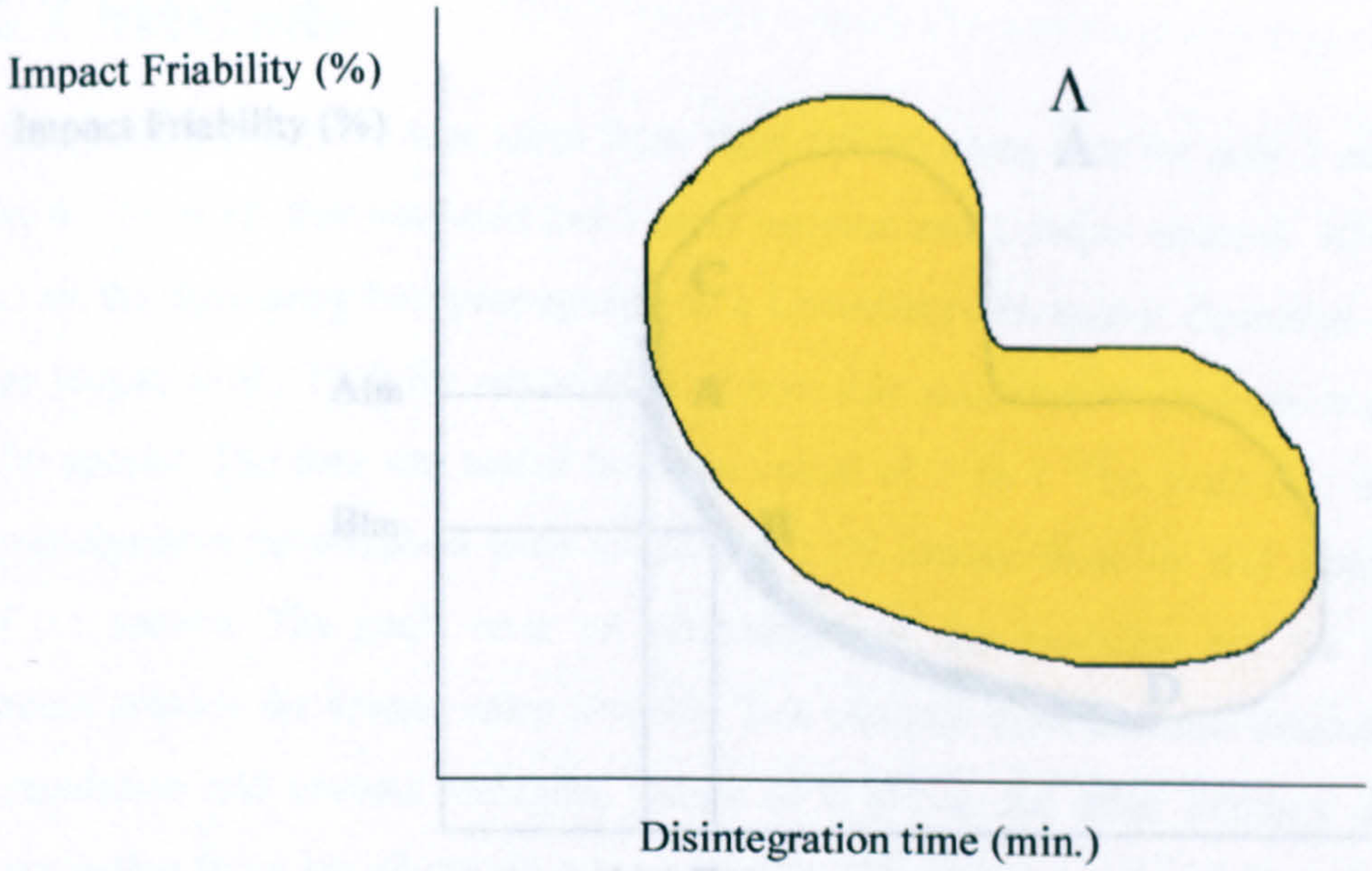


Figure 6.1: Objective function (F(x)) region (Λ) for disintegration time and impact friability. In the region, closed by the line (not the white area), are the allowed values generated from simulating ANN on independent variables - x (not true values, just for illustration purposes).

Since F(**x**) is a vector, the components of which (the two responses) are competing, there is no unique solution to the problem. The concept of non-inferiority (Zadeh, 1963), that is also called Pareto optimality (Censor, 1977), is used to characterise the objective function. A non-inferior solution is one in which improvement in one objective requires a deterioration of other one. In this case improvement of impact friability (lowering % friability) will cause deterioration in disintegration time (increasing disintegration time) and vice versa. From Figure 6.2, on the curve between points C and D there are two non-inferior points A and B. If one moves from point A to B there is a lowering of impact friability as **B**Im < **A**Im but also an increase in disintegration time since **B**di > **A**Di. Multiobjective optimisation is concerned with the selection of non-inferior solution points.

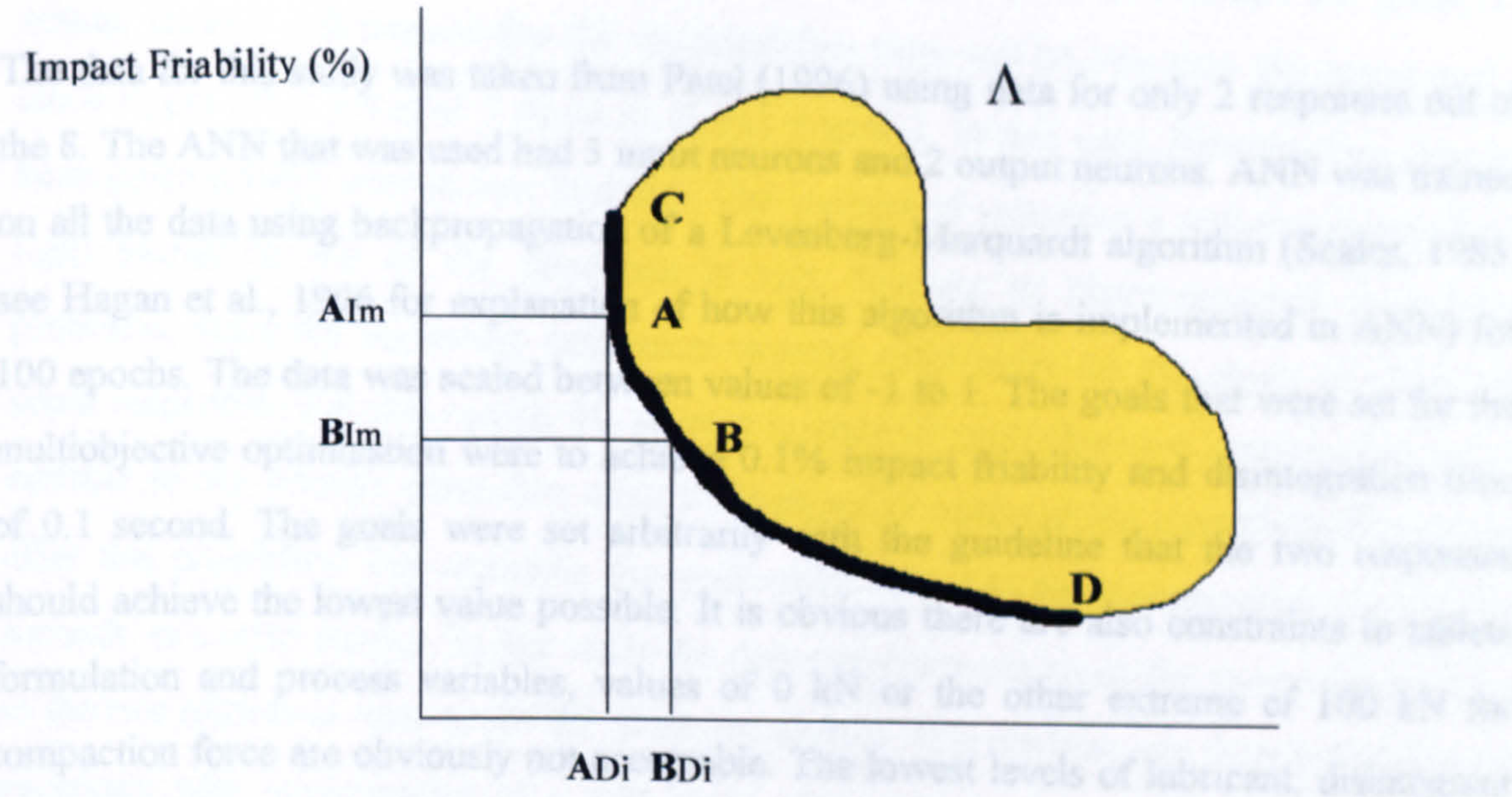


Figure 6.2: Selection of noninferior solution points. Moving from A to B causes improvement in impact friability response (reduction) but deterioration in disintegration time response (elongation). The region that is closed by the line (not the white area) is the objecive function space (Λ).

6.2 Methods

The data for this study was taken from Patel (1996) using data for only 2 responses out of the 8. The ANN that was used had 3 input neurons and 2 output neurons. ANN was trained on all the data using backpropagation of a Levenberg-Marquardt algorithm (Scales, 1985; see Hagan et al., 1996 for explanation of how this algorithm is implemented in ANN) for 100 epochs. The data was scaled between values of -1 to 1. The goals that were set for the multiobjective optimisation were to achieve 0.1% impact friability and disintegration time of 0.1 second. The goals were set arbitrarily with the guideline that the two responses should achieve the lowest value possible. It is obvious there are also constraints to tablets formulation and process variables, values of 0 kN or the other extreme of 100 kN for compaction force are obviously not acceptable. The lowest levels of lubricant, disintegrant and compaction force were 0.1%, 1% and 3 kN respectively. The upper levels were set to 100% for lubricant and disintegrant and 100 kN for compaction force. The starting values for the optimisation routine were 0.25%, 2%, 14 kN for lubricant, disintegrant and compaction force (case 9 in Table 3.1). Theses starting values were chosen since the input parameters yielded good response values of 0.91% and 21.67 seconds for impact friability and disintegration time respectively. Another starting guess was chosen by taking the input parameters of the first data point (see case 1 in Table 3.1).

The search for the best input parameters used the Goal Attainment method of Gembicki (1974). In applying this method to this study $F_1(\mathbf{x})$ is the ANN output with respect to impact friability and $F_2(\mathbf{x})$ is the ANN output with respect to disintegration time. The aim is to minimise γ (variable) subject to:

$\mathbf{x} \in \Omega$; $\quad F_1(\mathbf{x}) - w_1 * \gamma \leq \text{goal 1}$; $\quad F_2(\mathbf{x}) - w_2 * \gamma \leq \text{goal 2}$;

where w_1 and w_2 are the weights. These weighting vectors express a measure of the relative trade-off between the objectives. The first condition is obvious since it is not desirable that the optimisation routine searches for impossible values of $\mathbf{x}$ like concentrations not in the range 0 to 100%. The goal values define target responses that in this case were set to 0.1% impact friability for goal 1 and 0.1 second for disintegration time of goal 2. Obviously, these values are imaginary since in this system they are far too strict to be of any practical relevance. Hence, there will be under-attainment as opposed to over-attainment of the goals. Nevertheless, the goals set up the overall target and the place where the search

begins. The weighting factors of w_1 and w_2 set up the direction of the search from the goal point. By changing their values one can control the preferences between the goals, e.g. setting w_2 to 0 will impose that the disintegration time will be less than the target goal 2, since $F_2(x) - 0 * \gamma \leq$ goal 2 equal to: $F_2(x) \leq$ goal 2. In this case disintegration time has been given priority over impact friability. The concept of searching a region composed of a rigid barrier in one of the responses is very relevant to pharmaceutical formulation problems, since the regulatory authorities impose strict limits on the range of values for some responses. For this reason the current method of multiobjective optimisation seems suitable to our domain more than other multiobjective optimisation algorithms that do not offer this possibility. For example, the goal could be set to be disintegration time ≤ 600 seconds as a strict barrier. In this study the two weights were given the same values of 0.1 so the two responses were given the same importance. It is obvious just from looking at the equations why minimisation of γ will cause also minimisation of the responses; if the disintegration time equation is rearranged and the values used in this study are entered into the equation it takes the form:

$F_2(x) - 0.1 \leq 0.1 * \gamma$ which is $(F_2(x) - 0.1) / 0.1 \leq \gamma$

From the equation above, minimising γ imposes a more stringent condition on disintegration time, so the disintegration time has to be smaller. On the other hand increasing the weight will allow disintegration time to be bigger, so there will be more freedom to the condition and the goal will be less rigidly met. Figure 6.3 displays the actual parameters values, for the Goal Attainment method that were used in this study.

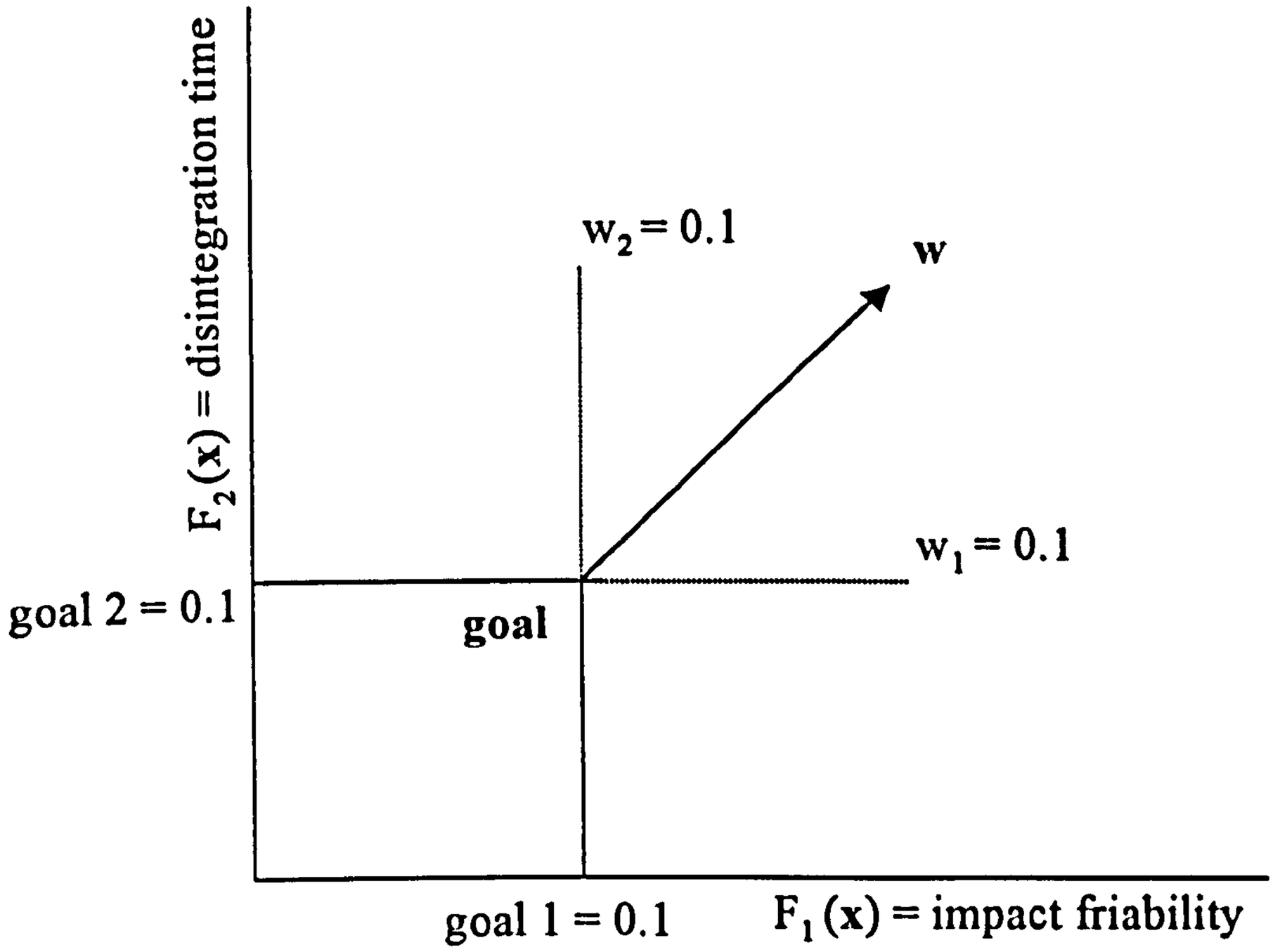


Figure 6.3: The parameter's settings used in this study for multiobjective optimisation using the Goal Attainment method.

The feasible function space Λ which is composed of $F_1(\mathbf{x})$ component on the x-axis and $F_2(\mathbf{x})$ component on the y-axis, no longer depends only on the independent variables constraints, but also depends on γ. Decreasing γ will cause a decrease in Λ, so the size of the feasible space is constantly changing during optimisation.

6.3 Results and discussion

After running the multiobjective optimisation program with the data of Patel (1996), the optimised solution was 0.1%, 3.12%, 16.9 kN for lubricant, disintegrant and compaction force respectively. The solution yielded the following responses values: 0.69% for impact friability and 8 seconds for disintegration time. Exactly the same solution was generated with the two different starting guesses, but the starting data point which was further away converges to the solution after 28 optimisation iterations as opposed to 33 iterations with the starting guess which was closer to the solution. This solution is better than any results generated in the experiments. Looking separately, at Table 3.2, for the best impact friability response one can see in case 17 a value of 0.65% (and 59 seconds for disintegration time) and for the best disintegration time the value in case 9 is 22 seconds (and 0.91% for impact friability). These response results are for two different input vectors of parameters as opposed to one input vector of the optimised solution. With respect to the disintegration time one can see the solution generated extrapolated beyond the response value, since the lowest disintegration time is 22 seconds

It is possible to set the goal of the optimisation routine to values that will not cause ANN to extrapolate beyond the response values. Setting the goal values to 0.65% and 21.67 seconds for the best impact friability and disintegration time response respectively yielded the corresponding results of 0.85% and 28.22 seconds for the impact friability and disintegration time respectively. The optimised input parameters for this result are: 0.1% 1.71%, 15.6 kN for lubricant, disintegrant and compaction force parameters respectively. However, the leave-one-out experiments that were described in previous chapters showed ANN can sometimes extrapolate to the response values quite reasonably. Thus, it may be unwise not to try a potentially good solution and to compromise on the second solution that is inferior because it uses more limited goals. Still, there is extrapolation problem with respect to both optimisation solutions, of the different goal settings, since they both suggested lubricant concentration of 0.1% and the lowest lubricant concentration used by Patel (1996) was 0.25%. To avoid this problem it is possible to set in advance the constraint of lubricant concentration to be not lower than 0.25%.

The optimisation path for the less stringent constraints (imposed by unreal goals) with relation to the response values is demonstrated in Figure 6.4. The first row of figures relates to the starting guess that is closer to the solution and second row is for the arbitrary starting guess. Looking at the graphs one could see that the optimisation routine has big fluctuations in the beginning and then the fluctuations became smaller and smaller as it approaches its solution. Looking at the top right plot one can see that at one stage of the optimisation the optimisation routine generated inferior disintegration time response than that of the starting guess due to very big fluctuations. The disintegration time after 5 iterations was 97.78 seconds, which is about five times greater than the starting guess. The point here, that one should not do too few optimisation iterations.

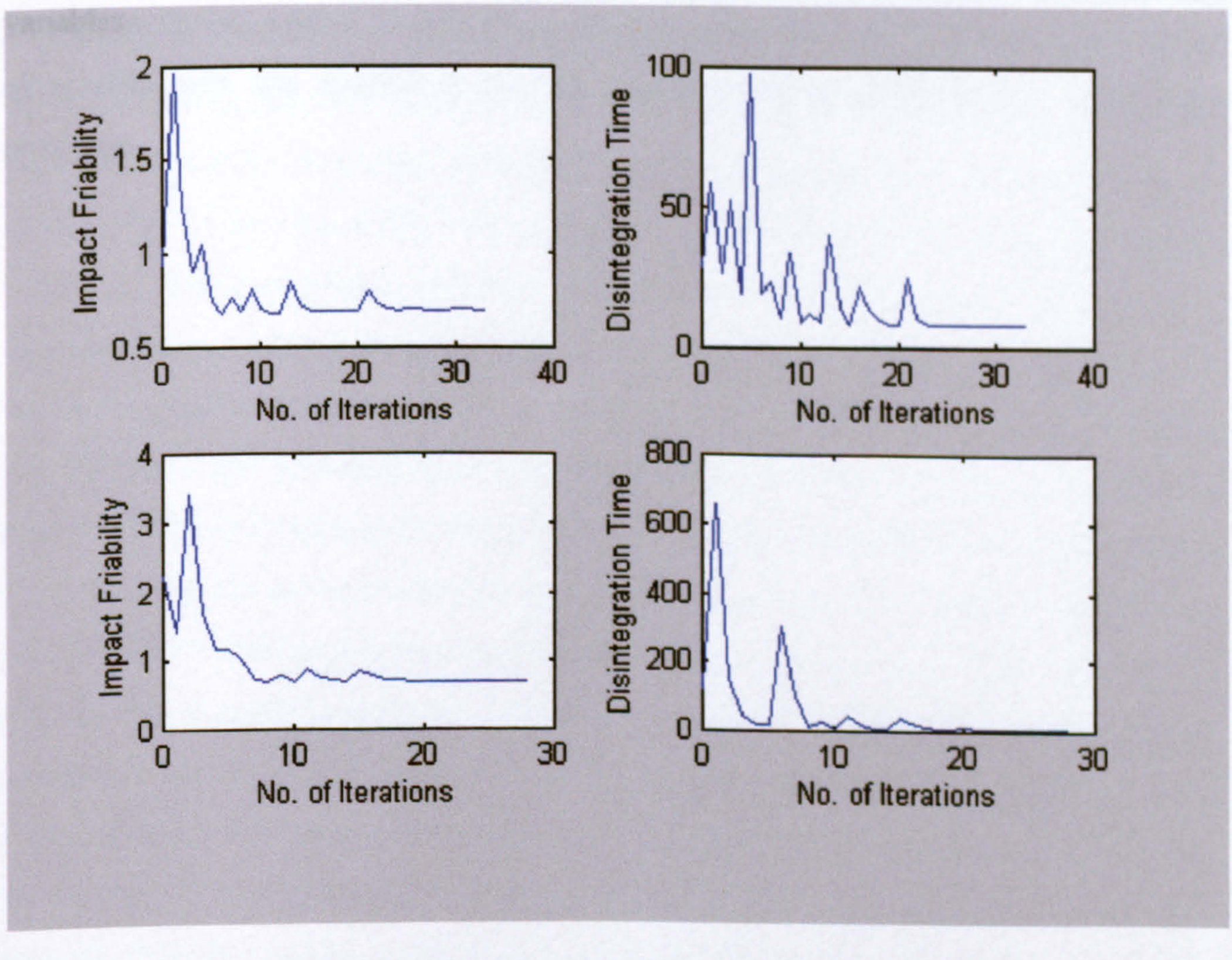


Figure 6.4: The simultaneous lowering of impact friability and disintegration time as a function of number of optimisation iterations, by using a novel computer program that incorporates ANN with the Goal Attainment method. The value of the response is on the y-axis. The number of iterations is on the x-axis. The first row of figures relates to the starting guess of the independent variables chosen to be close to the optimum solution (this starting guess was found by inspecting the experimental data). The second row relates to the arbitrary starting guess.

Multiobjective optimisation procedure forces one to have knowledge of the system to be optimised. The knowledge is important for telling the optimisation algorithm what are the optimisation goals and the relative trade-offs between the goals, but also to give it the constraints. For example, if from literature it is known that the amount of magnesium stearate should be 0.25 - 5% as lubricant, it must be defined as constraints to the optimisation algorithm. It is important to note that recommended concentration can change for the same excipient depending on its role in the formulation. If the constraints are not properly defined the optimum solution that is arrived at could have unrealistic values of independent variables. Hence, it is important that the expert system that incorporates the optimisation routine gives the user the knowledge necessary for optimisation, as in this case it is the recommended concentrations of the excipients that were defined as independent variables.

6.4 Conclusions

This study suggests there is no need to search for the data point with the best responses as the starting guess for multiobjective optimisation. In this study a random selection of data point out of the 27 tablet experiments arrived at the same optimum point but faster (with less optimisation iterations) than the data point with good response values that was especially selected. Another point is that one should not use too few optimisation iterations since there are big fluctuations in the beginning of the optimisation process, these became more moderate as the number of iterations increases.

This study investigated the use of multiobjective optimisation of a new method. It was feasible to do this method. It should be now more generally applied. Other people should experiment with this method in solving various practical multiobjective optimisation problems.

7. Expha Expert System

7.1 Introduction

A pharmaceutical formulator, retiring or leaving can cause a gap that impairs the ability of a pharmaceutical company to improve or create formulations for existing/new products. If the experiments that the formulator performed were not documented properly the problem is even greater. A new formulator would find it difficult to take the place of an experienced formulator and this problem is worse if he/she has no prior experience. The novice formulator does not know where to start and hence is likely to conduct more experiments than the experienced formulator, wasting the company's time and money. Another issue is that the formulation generated by a novice formulator will probably have inferior properties to one that the expert formulator would make, although both might pass the pharmacopoeia limits. The experienced formulator is more likely to arrive at the better formulation. Often, the problem is that experienced formulators fail to document fully their development process. This is partly due to the type of person pharmaceutical formulators sometimes are. Sometimes experienced formulators possess immense creativity at the expense of a less developed habit for documentation. In contrast, for example, one may say that quite the opposite is true for the quality assurance personnal. These statements are of course only generalisations that are not always true (based on my limited experience as a worker in a pharmaceutical company). As was mentioned before, poor documentation can impair the performance of a new formulator in a new position. In addition, the documentation of all available parameters is important for the success of optimisation process. A formulator conducting trial and error experiments can do many of these without arriving at an optimum formulation with adequate properties. Trial and error is necessary in many cases but the

data generated should be analysed rigorously, and sometimes, systematic studies may need to be done and only than one may arrive at optimum and robust formulation.

Not many pharmaceutical expert systems exist in the field of solid dosage form development. Cad/Chem (Colbourn & Rowe, 1996) is a system used for formulation optimisation not just in pharmacy. In the pharmaceutical solid dosage forms area it was used (Kesavan & Peck, 1995) to optimise tablet formulation. It uses ANN for building models and genetic algorithms for optimisation. It was mentioned first since it differs from the other expert systems that will be discussed. It has no knowledge of the problem and it is just a modelling and optimisation tool as opposed to the expert systems that will be discussed later that have knowledge of the pharmaceutical problem. There is an expert system for capsule filling and related problems in powder technology (Lai et al., 1996). This system consists of a database of excipients and a database of many formulations. It has very valuable knowledge of facts and rules generated from inquiring 10 experts in the field of capsule formulation development. There is an expert system for solving tablet's coating defects (Rowe & Upjohn 1993a). It was developed with the aid of a special tool for developing expert systems called Expert System Shell. Zeneca Expert System (Rowe & Upjohn 1993b) is used for solving tablet formulation problems. First it suggests a formulation, then it predicts the tablet properties of the suggested formulation. The user tries the formulation and 'feeds' the computer the observed values of the new tablet properties. Accordingly, the computer corrects itself and suggests another formulation. This iterative process continues till the user is satisfied from the tablet properties. The Cadilla expert system for tablet formulation (Ramani et al., 1992) suggests a formulation to the user, if the user rejects the system's advice it suggests another formulation. This latter system lacks an optimisation stage. An expert system for tablet formulation was developed using a multivariate statistical method called canonical analysis (Podczeck, 1992). It analyzed the behavior of different mixtures of excipients with 15 drug substances. The statistical analysis was translated into facts and rules.

Currently, there is no explicit write-up in the scientific literature of a tablet formulation expert system despite such software are available. Proprietary systems are very expensive so it was not possible to have access to them within the remit of this project. Nevertheless, there are general programs for formulation like Cad/Chem that could be used only for the formulation optimisation stage.

The purpose in creating Expha is to develop an expert system designed to aid the definition and optimization of a solid dosage formulation and manufacturing process. This will reduce the number of experiments. Hence, it shortens development time and saves money. The quality of formulation will improve since it will have better response properties. Another specification is to use the expert system to educate undergraduate, postgraduate and new scientists in the field of tablet formulation and to expose them to expert systems.

Typically (using Expha) one would collect the data, model this data as a function of the variables investigated using either regression or ANN and then carry out optimisation. There are models that are not likely to vary from one problem to another, these models could be incorporated within the decision making within Expha. Details of building models and optimisation are given in Chapter 2.

7.2 Specifications & features

Expha is an expert system designed only for tablet formulation. It should have a database of excipients and tools to allow expansion of this database by the user. It should also have a database of formulations and processes that will be created from scratch by the user. All this data acquisition will be done by a friendly user interface that will help to make the process of acquiring data an easy one and aid in preventing mistakes. Data retrieval from the databases should be an easy task that uses a friendly user interface. The requested data should be presented on the screen or in the form of reports. It should possess basic pharmaceutical knowledge in the form of rules. The rules are not meant to be comprehensive. Expha should have modelling and optimisation components. The modelling part will enable the user to choose between ANN and regression. In each one of the modelling methodologies the user should be able to build various models which may than be incorporated within Expha and used in decision making. The next section delves into specific issues concerning Expha.

Expha should give advice on the manufacturing process, formulation and miscellaneous issues like a recommendation to micronise a drug substance. Another goal is information collection, for the appropriate documentation of all the experiments with all the relevant parameters. For example, the precise documentation of process parameters and formulation ingredients is very important. The next issue is tightly connected to the previous one. It is

the use of the information gathered for developing regression/ANN models. The regression models could give insight into the pharmaceutical system; e.g. it can be seen in many pharmaceutical systems that as more lubricant is added dissolution rate deteriorates. The exact quantitative relation between lubricant level and dissolution rate could be seen from the regression equation. In a similar manner, in another system of a drug with different excipients other phenomena may be detected. The ability to look into the pharmaceutical system is an educating tool by itself that enhances the understanding of formulation and process issues. The novice formulator can see that a trend like the one described previously is repeated in many formulations and acquires more knowledge of formulation development, so taking into account the amount of lubricant for example. Another goal is the development and use of models for optimisation. After the experimental data is collected and models are created the software can be queried as to which parameters give a solution that will best satisfy the different responses.

For example, if lubricant or disintegrant levels were manipulated, what values of these parameters would give the golden path yielding good hardness and adequate dissolution rate. The ability to perform optimisation is cardinal since it reduces the number of experiments conducted, in turn reducing money spent on formulation development. The software generating optimal values by utilising efficiently every piece of information derived from experiments also improves the product quality. The next section will discuss the backbone of this expert system—its structure that will help it achieve its goals.

Expha expert system should be divided into three main parts, database, modelling and optimisation. There should be dependency between the parts since without data from the database models cannot be built and without models optimisation cannot be done. Expha's structure is presented in Figure 7.1.

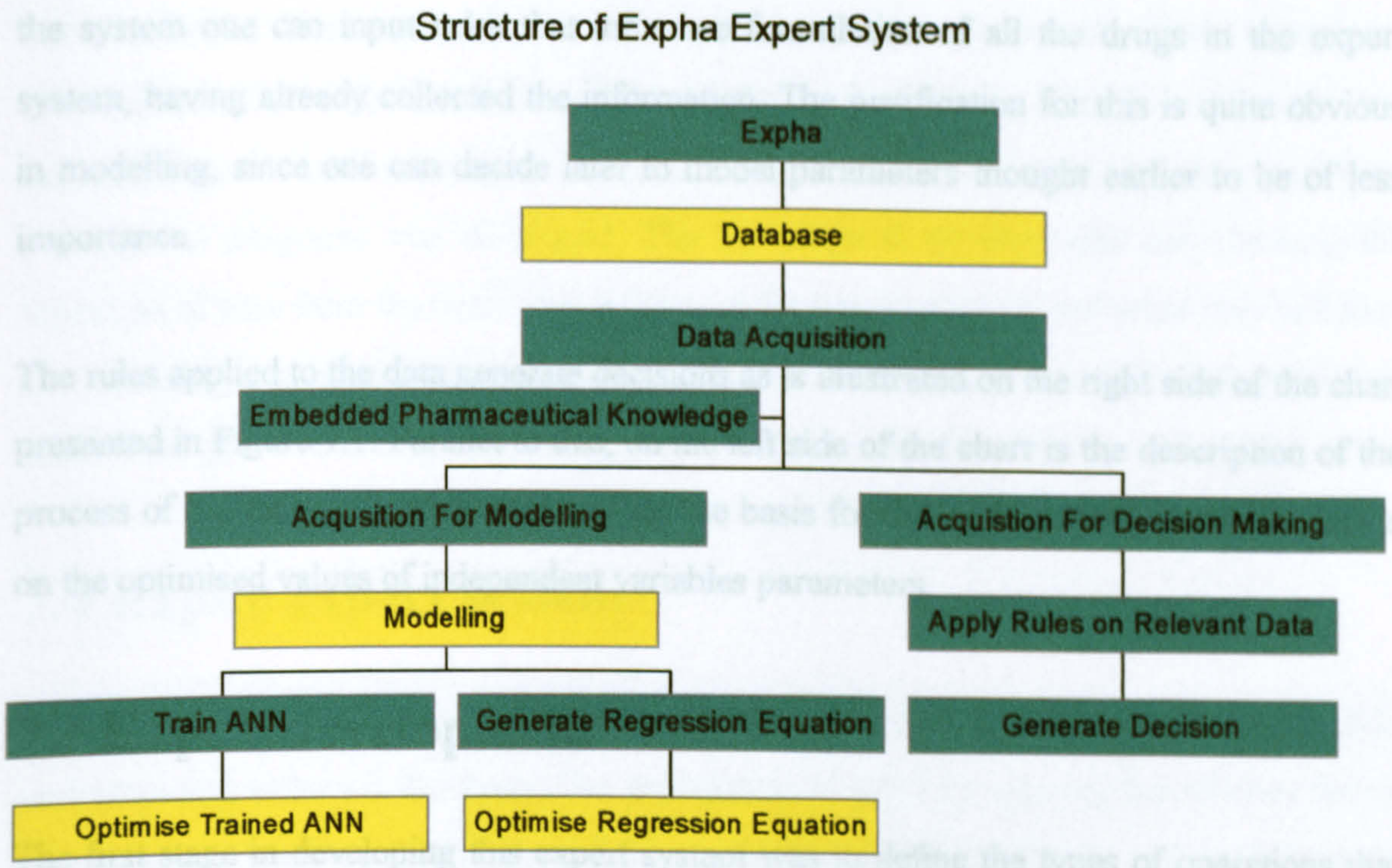


Figure 7.1: The structure of Expha. The yellow boxes represent the main parts of Expha.

The yellow boxes in Figure 7.1 represent the main parts of the Expha. The database part should be divided into two parts, data acquisition for decision-making and data acquisition for modelling. Decision-making means to decide on the process, warns the user when he selects an excipient quantity that is not within the recommended concentration range, give formulation and process tips to the user etc. In all these acts the software responds according to the data in the database. The program designer influences the software's method of data acquisition from the user. The embedded pharmaceutical knowledge of the database developer influences the collection of data also for modelling, e.g. inputting fields which seem to be relevant for describing the drug entity and selecting process variables which seems to be relevant, and this can influence the modelling. This is the developer's subjective concept that is derived from his pharmaceutical knowledge. Pharmaceutical knowledge is also influential in deciding on the rules applied to the data. Not all the data that will be acquired is used for decision making or for modelling. The reason for putting the extra data will be explained by examples; the "Drug properties" form should include properties that seem important for designing formulation. Part of these fields will be used only for data storage and for viewing. Why is it necessary to store data in fields if no operation is performed on them? There are two reasons, sometimes seeing relevant data can influence decisions (regarding formulation or process) and it might be that in the future some rules would be applied to these fields. Once one has the data on numerous drugs in

the system one can input rules that influence formulation of all the drugs in the expert system, having already collected the information. The justification for this is quite obvious in modelling, since one can decide later to model parameters thought earlier to be of less importance.

The rules applied to the data generate decisions as is illustrated on the right side of the chart presented in Figure 7.1. Parallel to this, on the left side of the chart is the description of the process of building up models that will set the basis for the optimisation routine to decide on the optimised values of independent variables parameters.

7.3 Expha development

The first stage in developing this expert system was to define the types of operations that would be performed. Two major activities were defined, one is numerical in nature and the second is linked to acquisition and manipulation of data. Another aspect that is technical in its nature and less of a programming challenge is the user interface. Although it is probably the easiest part to develop this may be the most important part. It was necessary to find a computer language that is strong in this field. From analysing these activities it was decided to use a computer language with powerful calculation ability on a platform based on a database. Hence, the base for the core of the system was the MATLAB® programming language for the numerical calculations and Microsoft Access® for dealing with data. Visual Basic® for Applications was chosen for the user interface language since it is very fast and efficient for this purpose.

The next step was to build a program with MATLAB® that would perform all the necessary operations of model building and optimisation. It was followed by the step of analysing the types of data that would be used. After all data types were characterised it was necessary to determine how the data tables relate to each other. Otherwise data will *float* in the database without the ability to be retrieved in the form of a report or any other relevant manner. After the overview of the data layout is clearly determined it is necessary to delve into the details of each data table. For example, it is imperative to define all drug properties that are of interest for formulation development. Someone who knows a lot about tablet formulation development or at least has extensive background knowledge about this subject does this. After the database was constructed the data manipulation algorithms

were developed. That means calculations on the data or data queries etc. These algorithms were implemented with Visual Basic® for Applications. After the database and numerical calculation parts were developed, the mechanism of communication and data exchange between the programs was developed. The last stage in development was planning the collection of data from the user. This influences the types of forms and what they will look like. Building a graphical user interface (GUI) is a methodology on its own that helps to achieve the goal of building a user interface that is truly user friendly.

7.4 Expha expert system

The aim of this section is giving the user ability to use the Expha software with a better understanding of how it functions. The explanation is aided by viewing the relevant forms. The emphasis is on a small number of forms found within the software and explaining the logic behind them. Some of the forms presented will be the main ones and others represent the group of forms they are derived from.

First will be given a general overview of this expert system. Afterward comes the use of the software. The guide through using the software follows the natural flow of the program with the collection of data and building formulations. Then model building and optimisation is discussed, as it is the natural continuation of using the data that has been collected. The section ends with the on-going process of maintaining the excipients database.

7.4.1 General overview

In this section a general overview of Expha will be undertaken by examining the menus. Looking at Figure 7.2, there are general Windows® menus; File, Edit on the left side and Window, Help on the right side. The five menus in between were created especially for Expha and will be discussed. From the left, "Excipients Expert" menu is composed of forms that enable maintenance of an excipients database. "Tablet Expert" menu is composed of forms that make up the backbone of the program. The data collected on these forms can be used for building models and for optimisation. This menu includes the database engine that executes the algorithm, which decides on the process and also functions that give tips to the user regarding process and formulation. "Raw Data" menu includes the tables of data which were collected with the forms that belong to the "Tablet

Expert" menu. These tables, of raw data menus, are used by copying the relevant data within them into Microsoft Excel® sheets embedded in a form ("Data Input") under the "Optimisation" menu, because this form includes the facilities to input the data for building models and optimisation. The rest of the forms under "Optimisation" menu" define the type of regression or ANN models and is used to set optimisation parameters for regression/ANN. The fact that the "Optimisation" menu includes programs that are computationally intensive means the speed of running these programs is dependent upon parameters like speed of the processor and amount of RAM memory. "Example Formulation" menu includes forms that give formulations examples according to the process.

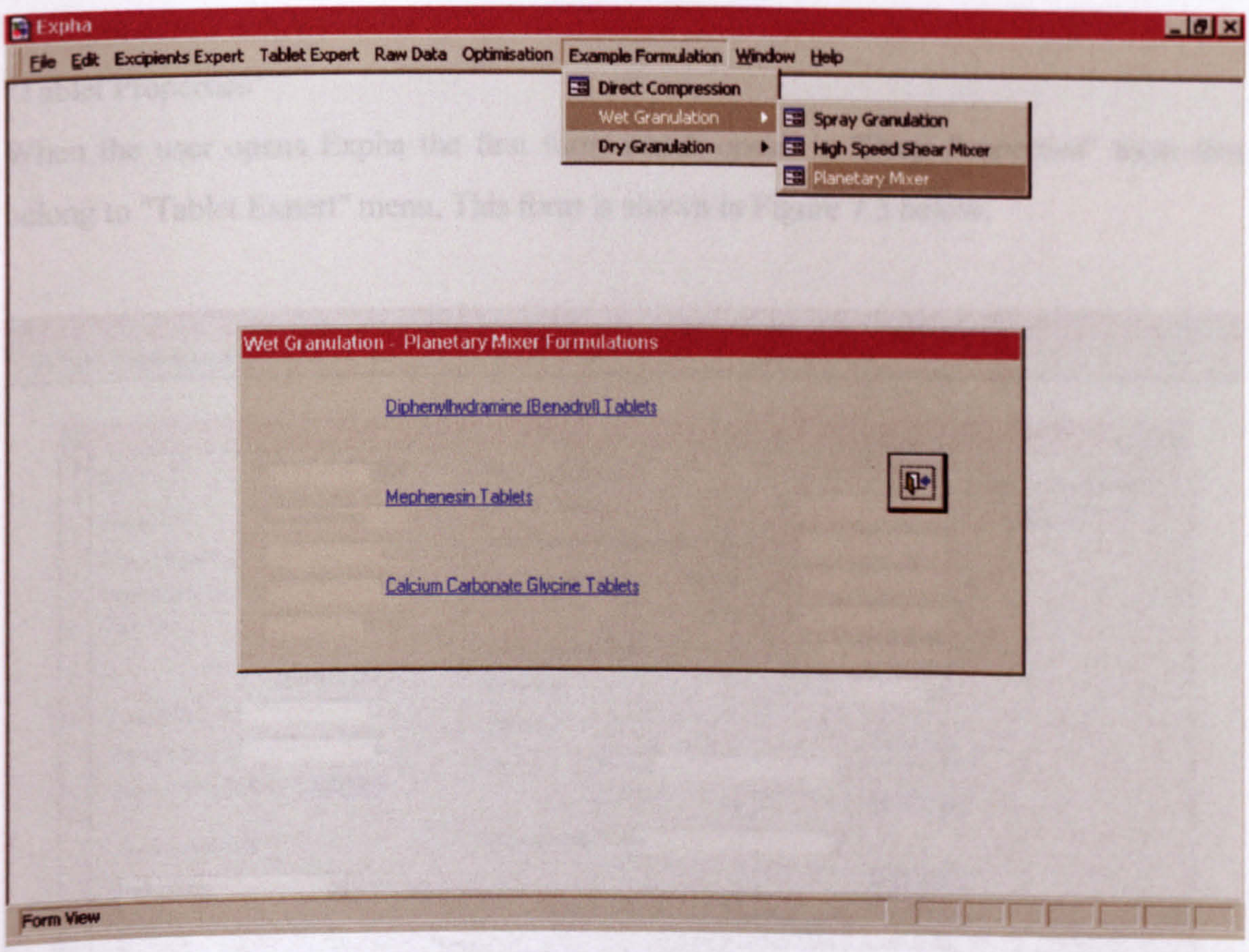


Figure 7.2: One form out of example formulation part.

In the example in Figure 7.2, the user has chosen from "Example Formulation" menu the "Wet Granulation" sub-menu and from this a planetary mixer was chosen. The end product is a form called "Wet Granulation - Planetary Mixer Formulations" with 3 hyperlinks that will each open the relevant formulation according to the name of the hyperlink when selected. Later in this chapter, only the parts that relate to the other four menus "Excipients Expert", "Tablet Expert", "Raw Data" and "Optimisation" will be discussed.

7.4.2 Flowing with the program - "Tablet Expert" menu forms

In this subsection Expha flow is explained using the "Tablet Expert" menu forms. "Tablet expert" menu forms includes the following forms:

"Drug Properties"

"Identify Process"

"Select Excipients"

"Process Variables" (with several subforms)

"Granule Properties"

"Compression Press Variables"

"Tablet Properties"

When the user opens Expha the first form which opens is "Drug Properties" form that belong to "Tablet Expert" menu. This form is shown in Figure 7.3 below.

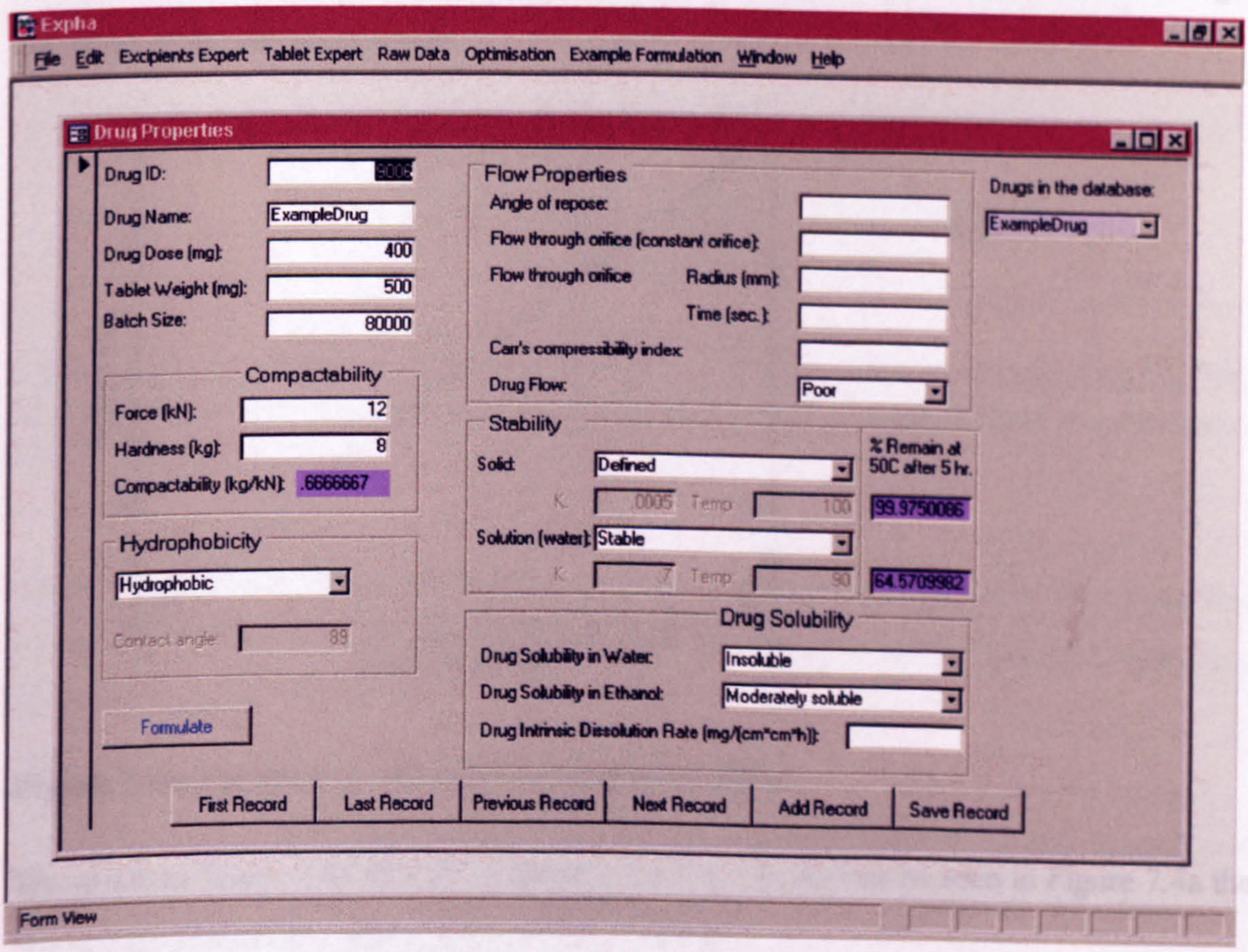


Figure 7.3: "Drug Properties" form of Expha expert system.

"Drug Properties" form is composed of different fields, describing the drug. Drug properties that were considered important and were put in the form are compactability, hydrophobicity, flow properties, stability and solubility. According to the values entered by the user, Expha recommends a process, computes values online and gives formulation tips. The button "Formulate" transfers the user to the next form, which actually recommends the process.

Understanding how to fill data in this form is a very important step in understanding Expha. Beginning with the first field on the upper left the user gives drug identity number ("Drug ID"), then the name of the drug, in this case called "ExampleDrug". Afterwards, the drug dose, tablet weight and batch size is filled in. These fields are important not just to this form but also to the form used to select the formulation. The other fields describing drug properties are discussed below.

Compactability value is calculated by dividing the hardness with compaction force to get the slope of the following graph:

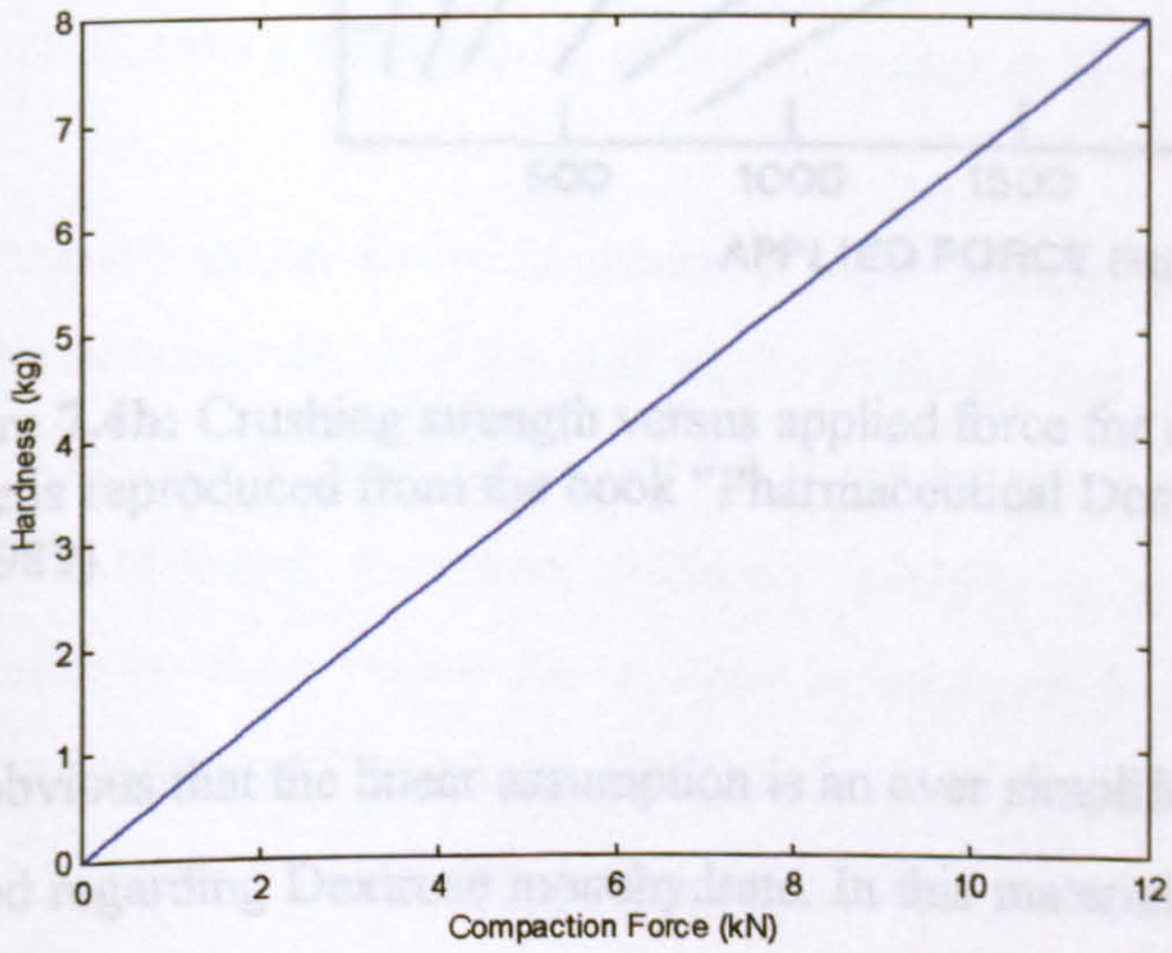


Figure 7.4a: Hardness as a function of compaction force.

The graph in Figure 7.4a follows equation $y = a * x + b$. As can be seen in Figure 7.4a the line passes through the origin, so $b = 0$ and $y = a * x$.

The slope is $a = y / x$, and the calculated graph slope is $a = 8 / 12 = 0.667$. This graph is a gross approximation of the behaviour of materials, this limitation is quite clear since the pattern of this graph will change with different materials. Also, for the same material the

graph may not be linear over all the range of compaction force. Compactability value of less then 0.5 was set as the value under which the drug cannot be directly compressed, but instead a granulation process should be used. Figure 7.4b taken from the book "Pharmaceutical Dosage Forms Tablets" (Lieberman et al., 1989) shows real examples of how various materials behave under increasing compaction force. It represents the variation of crushing strength with applied force.

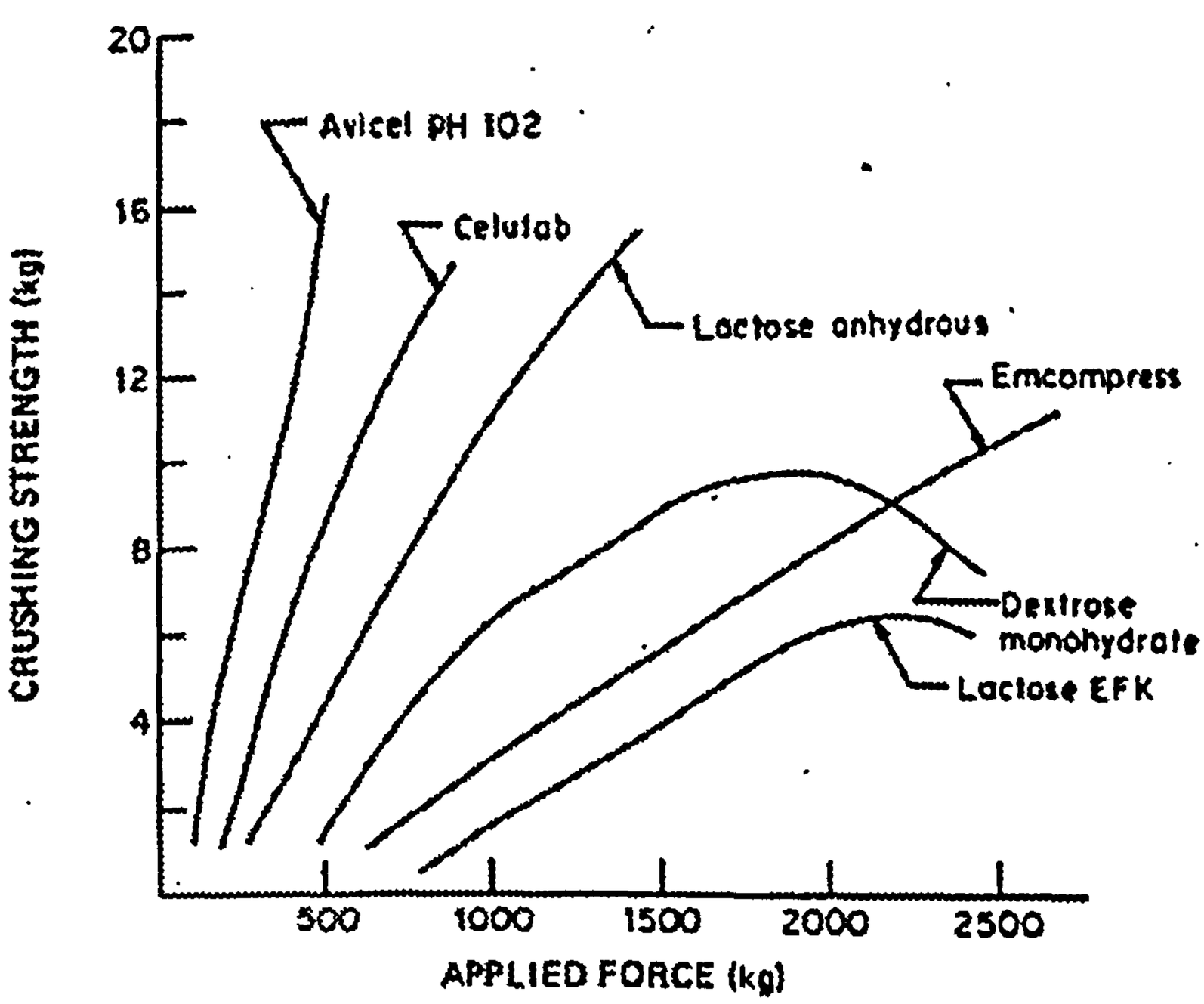


Figure 7.4b: Crushing strength versus applied force for compacts of various materials. The figure is reproduced from the book "Pharmaceutical Dosage Forms Tablets" (Lieberman et al., 1989).

It is obvious that the linear assumption is an over simplified one as could be seen in the line plotted regarding Dextrose monohydrate. In this material, the linear behaviour is valid only for the low compaction force.

The next group relates to drug hydrophobicity. The user can fill literal values, or to choose "Defined" from the combo box (combo box is a type of field that enables the selection of certain values out of a predefined list) thus enabling the user to enter a contact angle value. The hydrophobicity value is worth considering when developing the formulation since it influences dissolution rate and success in tablet coating. To simplify, the less hydrophobic a drug is the more chance it will dissolve faster. The angle which the liquid makes with the

solid surface at the point of contact is the contact angle (Gennaro, 1990). When wetting is complete the contact angle is zero and when there is no wetting the contact angle is 180°. In the latter instance the drop touches the surface at only one point. It is wise to measure contact angle with water since the industry is moving toward aqueous tablet coating to prevent environmental damage and explosion hazards of organic liquids. The lower the contact angle of water with drug is, the more chance there is that coating the drug with aqueous solution would yield better adherence to the surface of drug (in that case the tablet surface). Contact angle is also important in granulation in assessing whether the binder solution will stick to the granules.

There are various fields to fill for the Flow Properties. Selection of a literal value from "Drug Flow" combo box is important for the decision on the process and the formulation. Currently the field's numerical values are not incorporated into the decision making of the process. Nevertheless, they can be used by setting numerical values as cut off zones which define what is poor flow for each field, e.g. user enters some value x for angle of repose (refer to section 2.2 for description of angle of repose), if x is bigger then y, which is the cut off value, then the flow is poor. Limitation to this is that studies showed that angle of repose by itself is not a good predictor of flow (Amidon & Houghton).

The stability group (refer to section 2.2 for discussion on stability), in contrast, already incorporates into the decision making the literal or numerical values since the cut off zone is set to a numerical value. In Figure 7.3 it can be seen in the solid stability combo box that the user choosing the value "defined" enables k and Temp. fields (k for a given temperature—these values can be found in books of drug profiles, e.g. "Analytical Profiles of Drug Substances" by Florey, 1982) and allows the use of the appropriate values entered. Just as in the compactability group, the result is calculated automatically by the software. The value calculated is the percent of drug remaining after 5 hours at 50°C (These values were set as an approximation to conditions in the granulation process). Expha calculates this value by taking into account two assumptions, explained with a numerical example below. The first assumption is that by rule of thumb, every 10°C increase in temperature causes k, which is the rate constant of decomposition, to double (Lachman et al., 1986). The value is 100°C in the temperature field (Figure 7.3) so k will be consecutively halved 5 times because 100 - 50 = 50. In other words, for k_{new} (the k for 50°C) k will be divided by

50°C) k will be divided by ten to get a value of $5*10^{-5}$ hr^{-1}. The second assumption is that the drug decomposes according to first-order kinetics as if it were in aqueous solution. Hence, the following equation (Martin et al., 1983) becomes applicable:

$$\text{Log}\,(C_{in} / C_{out}) = t * k_{new} / 2.303$$

Whereas C_{in} is the initial drug concentration entering the granulation process, C_{out} is the final drug concentration going out from the granulation process, in this case after t = 5 hours. Putting the user values:

$$\text{Log}\,(C_{in} / C_{out}) = 5*10^{-5} / 2.303$$

$$C_{in} / C_{out} = 10^{1.0855E-4}$$

And the percent remaining is $C_{in} / C_{out} * 100 = 99.975\%$ which is the result in Figure 7.3.

Regarding stability in solution (water) property (Figure 7.3), the user chooses a literal value of "Stable". Alternatively, the user can selects numerical values for k and Temp. variables. The cut-off zone for allowing wet granulation is 95%, so below that value of percentage drug remaining Expha will not recommend wet granulation. The assumption here is that all the granulation process is done when the drug is dissolved in water, in reality the drug is in a semi-solid state. This is why allowing a high value of 5 percent decomposition compensates for the too rigid assumption. It can be seen in Figure 7.3 that since a literal value was chosen for stability in water, the k and Temp. fields became disabled.

The final group that characterises drug contains fields that relate to drug solubility. The solubility values of the drug are very important for the decision on the formulation. One of the important responses they influence is the dissolution of tablet. If drug solubility in water value is low, operations like adding surfactant or micronisation of drug should be considered. If the solubility value is high and the drug is high dose, it is possible that a disintegrant may not be needed.

After the drug properties form is filled the user can use the formulate button which runs the following algorithm (Figure 7.5) to recommend on the process:

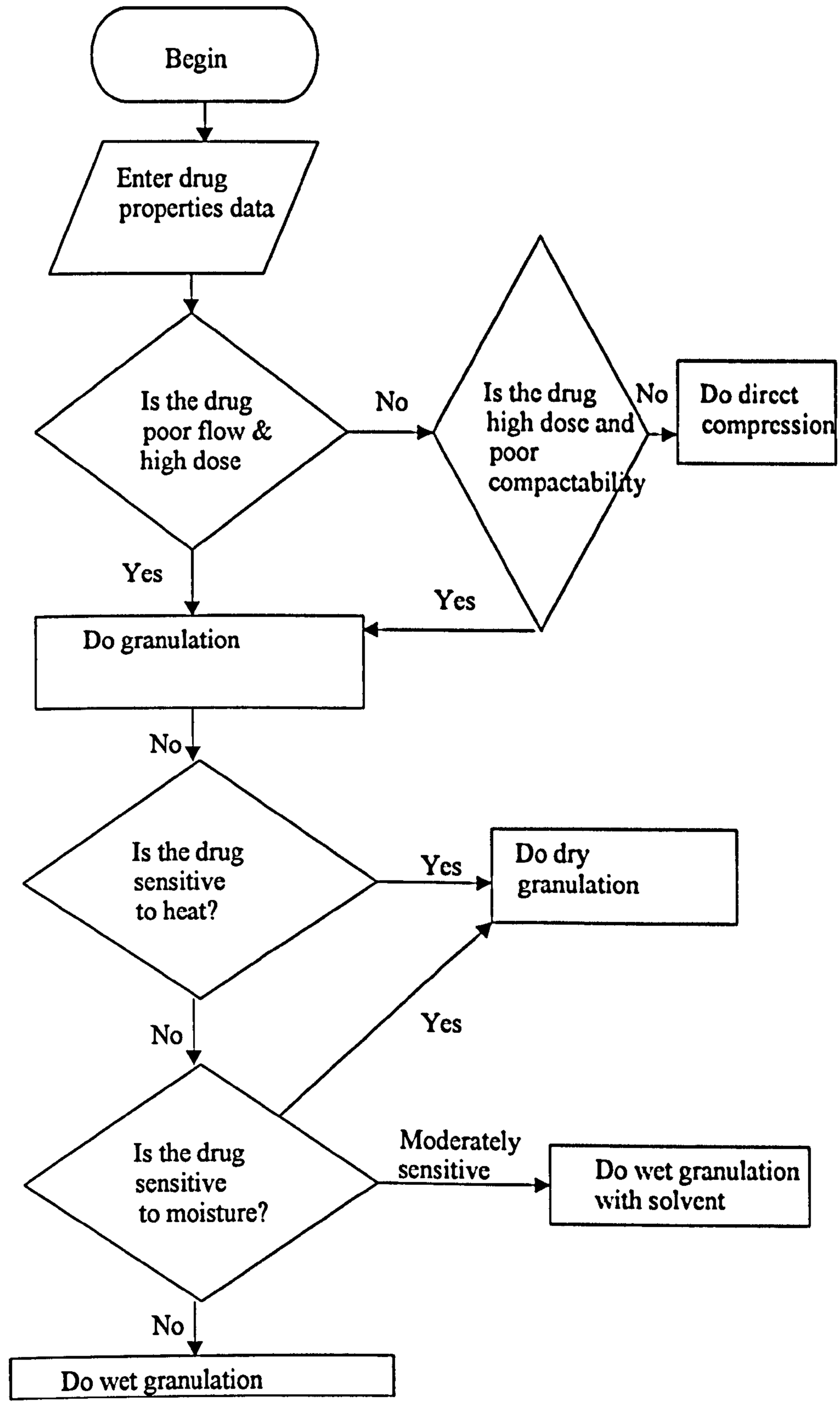


Figure 7.5: Expha algorithm for process decision.

What is missing in this flowchart is where the boundaries (cut-off zones) lie for each decision, some of these values were mentioned earlier. Generally it was decided to use reasonable values, e.g. ≥ 80% of active drug was defined as high dose. The execution result of the flowchart is seen in the "Identify Process" form (Figure 7.6), which follows the "Drug Properties" form. Expha automatically chooses the process by selecting the appropriate button on the "Identify Process" form. There is also an explanation of the selection. However, the user can reject Expha recommendation and select another process.

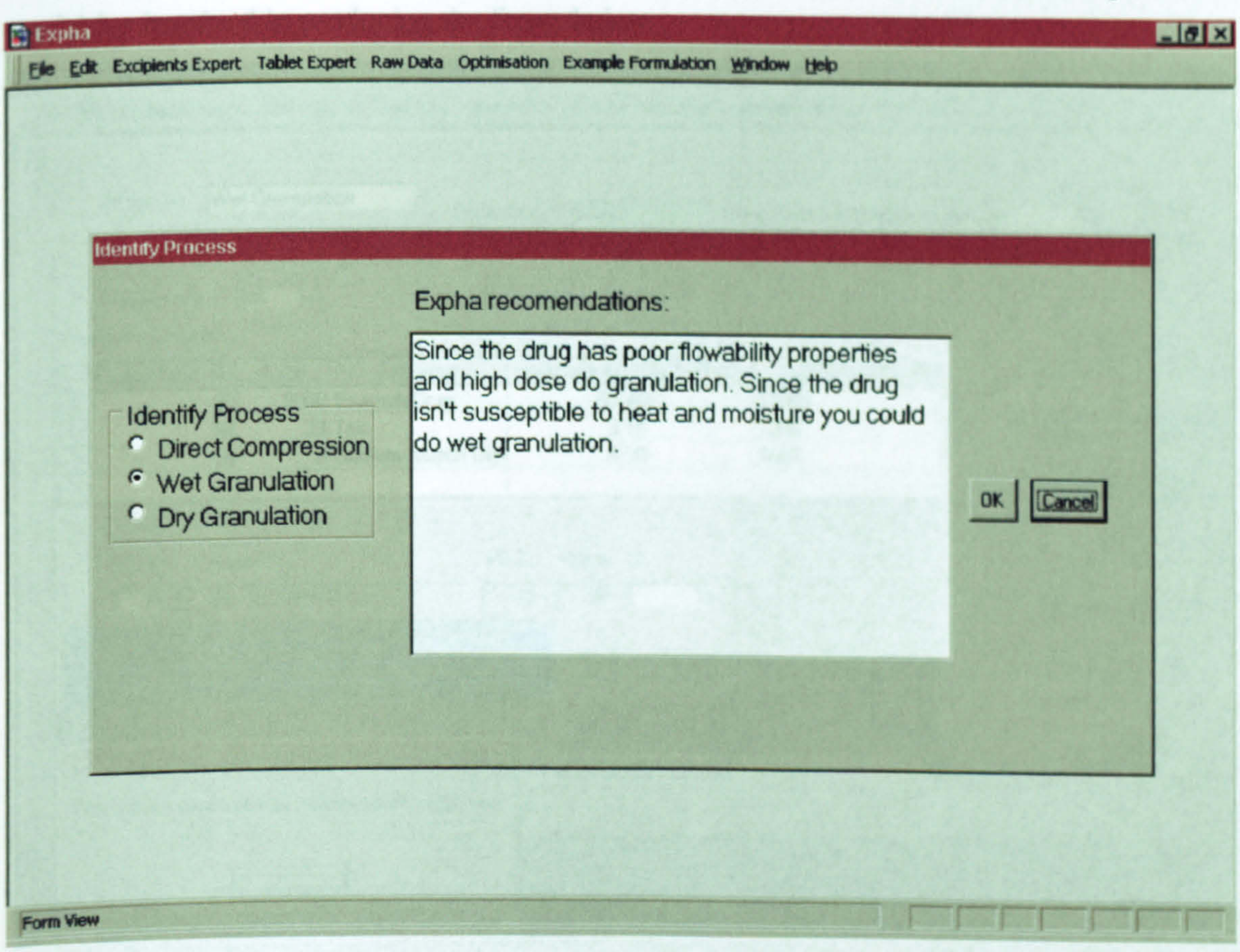


Figure 7.6: "Identify Process" form

In this specific example the recommendation was arrived using the straight-line path in the algorithm (q.v. Figure 7.5). It responds to the relevant fields in the "Drug Properties" form, shown on Figure 7.3. In this example, after Expha decided the process would be granulation, it took into account the solid stability field, which represents susceptibility to heat, and stability in solution (water) field as a measure of susceptibility to moisture. From Expha recommendation in Figure 7.6 it can be seen that the latter two fields on the "Drug Properties" form were taken into account to arrive at wet granulation as the recommended granulation process.

Looking at "Select Excipients" form in Figure 7.7, all the fields in colour are data taken from the "Drug Properties" form and the user cannot change them through "Select Excipients" form, but has to revert to the "Drug Properties" form. The Process field was taken from the "Identify Process" form and can be changed just by returning to this form. All the user does in this form is to give a unique number to "Formulation ID" and to choose the formulation ingredients. "Formulation ID" is a number, which is connected to formulation ingredients and also to all data relevant to the formulation, like the process variables involved in producing the formulation.

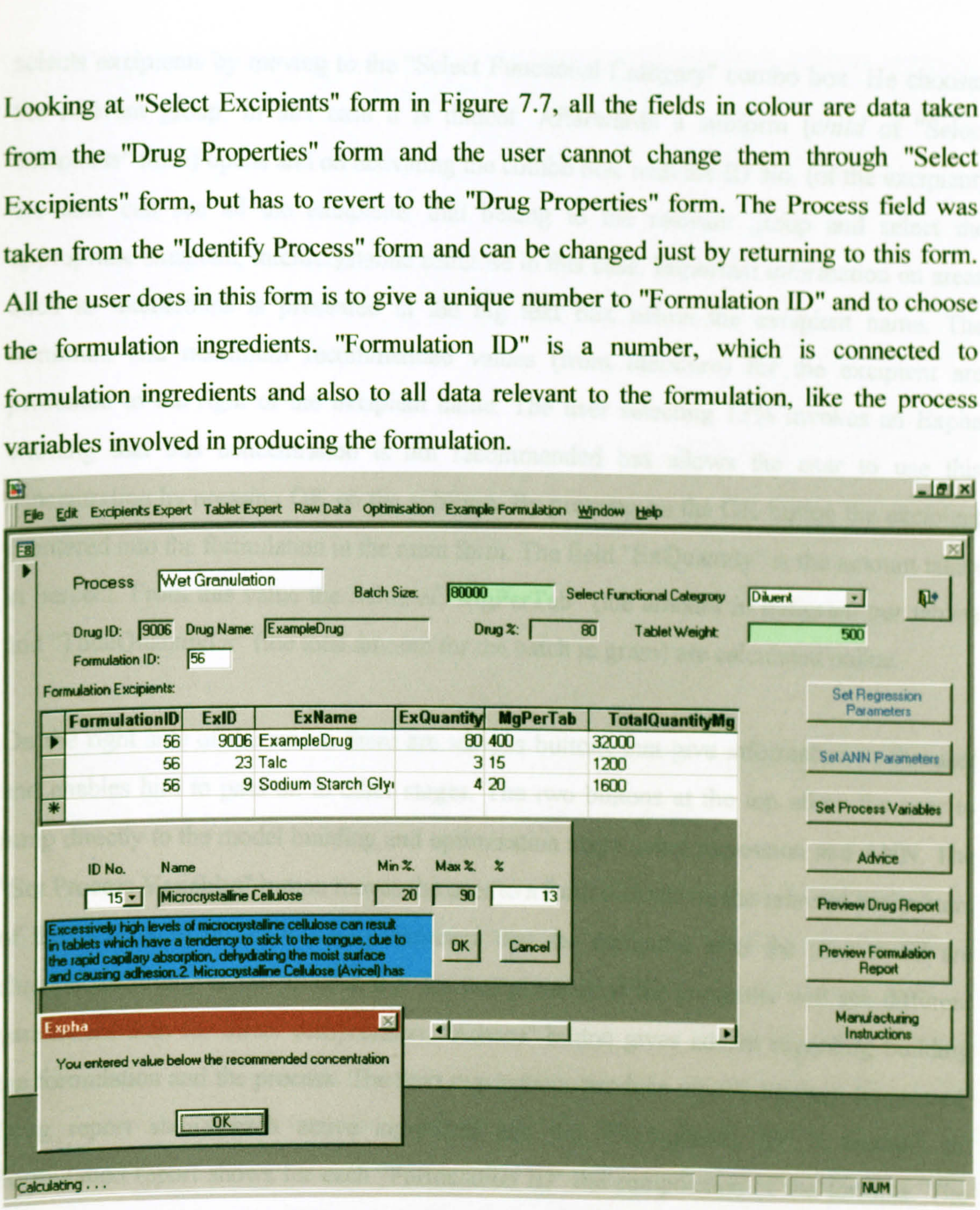


Figure 7.7: "Select Excipients" form.

To use this form first a value is given to "Formulation ID" field, in this case 56. Expha will not allow the user to first select the excipients thus avoiding situations where the user has built up a formulation and forgotten to link it to a "Formulation ID". Otherwise the formulation data would sit uselessly in the database with no possibility of use. Upon entering "Formulation ID", if there is already a formulation with this ID number it will be shown immediately on the screen. So this screen enables the user to enter new formulations as well as updating and looking at old ones. After entering "Formulation ID" the user selects excipients by moving to the "Select Functional Category" combo box. He chooses

selects excipients by moving to the "Select Functional Category" combo box. He chooses the relevant group; in this case it is diluent. Afterwards a subform (*child* of "Select excipients" form) opens and on activating the combo box with the ID No. (of the excipient) the user can see all the excipients that belong to the relevant group and select the appropriate excipient, microcrystaline cellulose in this case. Important information on areas such as interactions is presented in the big text box below the excipient name. The minimum and maximum recommended values (from literature) for the excipient are presented to the right of the excipient name. The user selecting 13% invokes an Expha warning that this concentration is not recommended but allows the user to use this concentration by pressing OK on the subform. By pressing on the OK button the excipient is entered into the formulation in the main form. The field "ExQuantity" is the amount taken in percent. From this value the fields of "MgPerTab" (the amount in miligram per tablet) and "TotalQuantityGr" (the total amount for the batch in gram) are calculated online.

On the right side of the screen there are various buttons that give information to the user and enables him to pass on to other stages. The two buttons at the top allow the user to jump directly to the model building and optimisation stage using regression and ANN. The "Set Process Variables" button moves the user to a form that allows the relevant parameters of the process to be input. The parameters that the computer asks the user to fill are changed according to the process, e.g. for wet granulation the computer will ask different parameters than for direct compression. "Advice" button gives advice regarding building the formulation and the process. The next two buttons produce reports that can be printed. Drug report shows each active ingredient and the "Formulation ID" it belongs to. Formulation report shows for each "Formulation ID" the composition of the formula. The "Manufacturing instructions" button gives manufacturing instructions that are changed according to the process.

Going back to our example, after selecting microcrystalline cellulose as diluent at 13% the value is entered into the formulation which now adds up to 100% of all formulation constituents. When the user pushes the "Advice" button, the screen looks as in Figure 7.8:

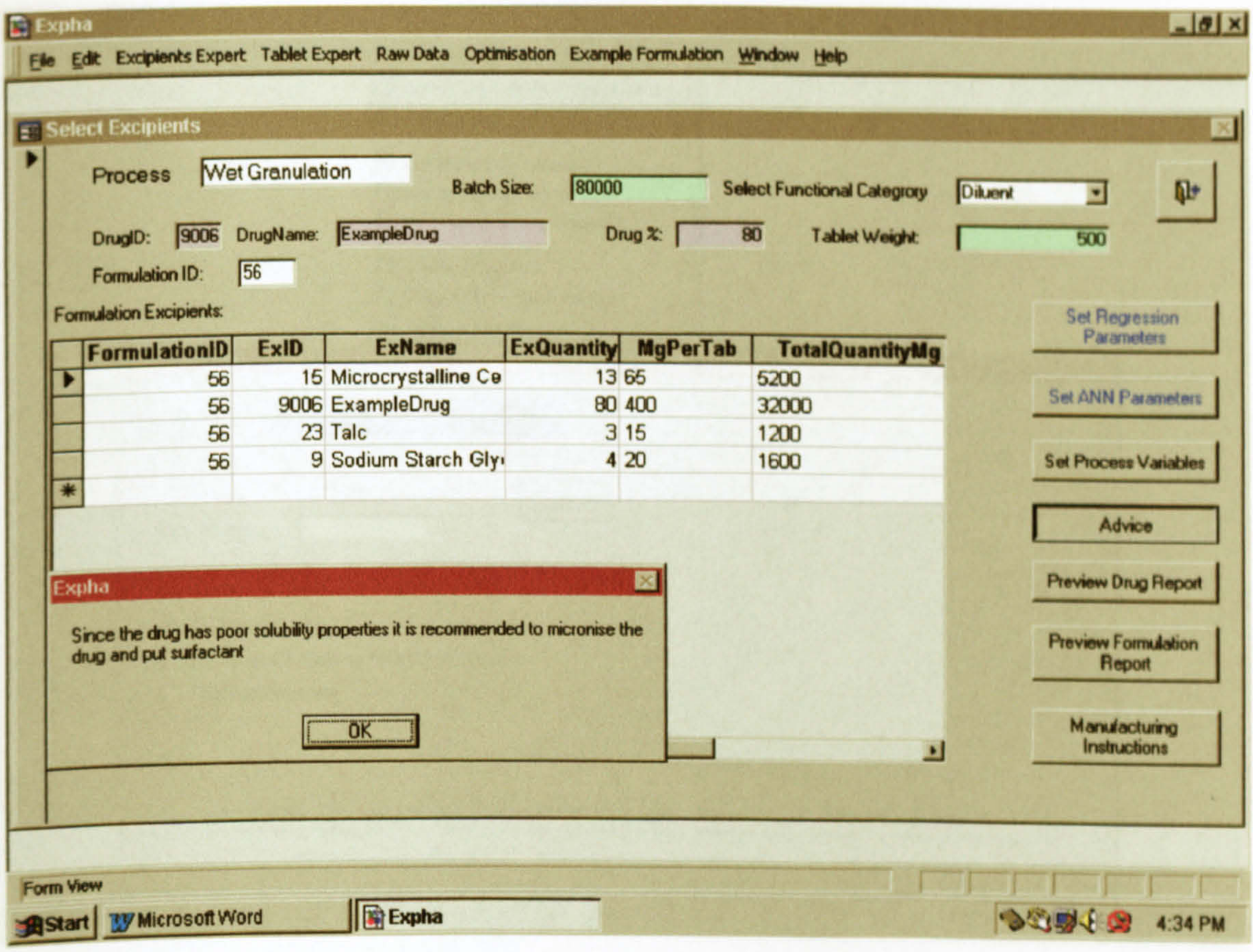


Figure 7.8: "Select Excipients" form with invoked advice remark.

The remark was generated in response to values entered into the database relevant to the virtual drug named "ExampleDrug". "Set Process Variables" button opens the form in Figure 7.9, which asks the user for information about the process. To add a new record the user selects the "Add Record" and "Formulation ID" value is copied from select excipient screen as the default value. Hence, after filling this form there are two sets of data, which relate to the same formulation (see "Formulation ID" field in Figure 7.8) that is 56 in this case. One set is the formulation ingredients and the other one is process variables. In this manner data about the same formulation ID is collected using the subsequent forms, pressing "Continue" to move from one form to another according to Expha flow. The final form is the "Tablet Properties" form, which includes the different responses like disintegration time, dissolution rate etc. At the end of the collection data process there are several sets of data, each one stored in a different table. This type of database structure is relational and not flat since it consists of several tables that are related to each other. This relational database structure enables all the data relevant to the same formulation ID to be retrieved. Figure 7.9 shows the wet granulation process variables form, and the "Raw Data" menu from within the relevant sets of data (generated by filling the forms) can be put into "Data Input" form as data for building models and optimisation.

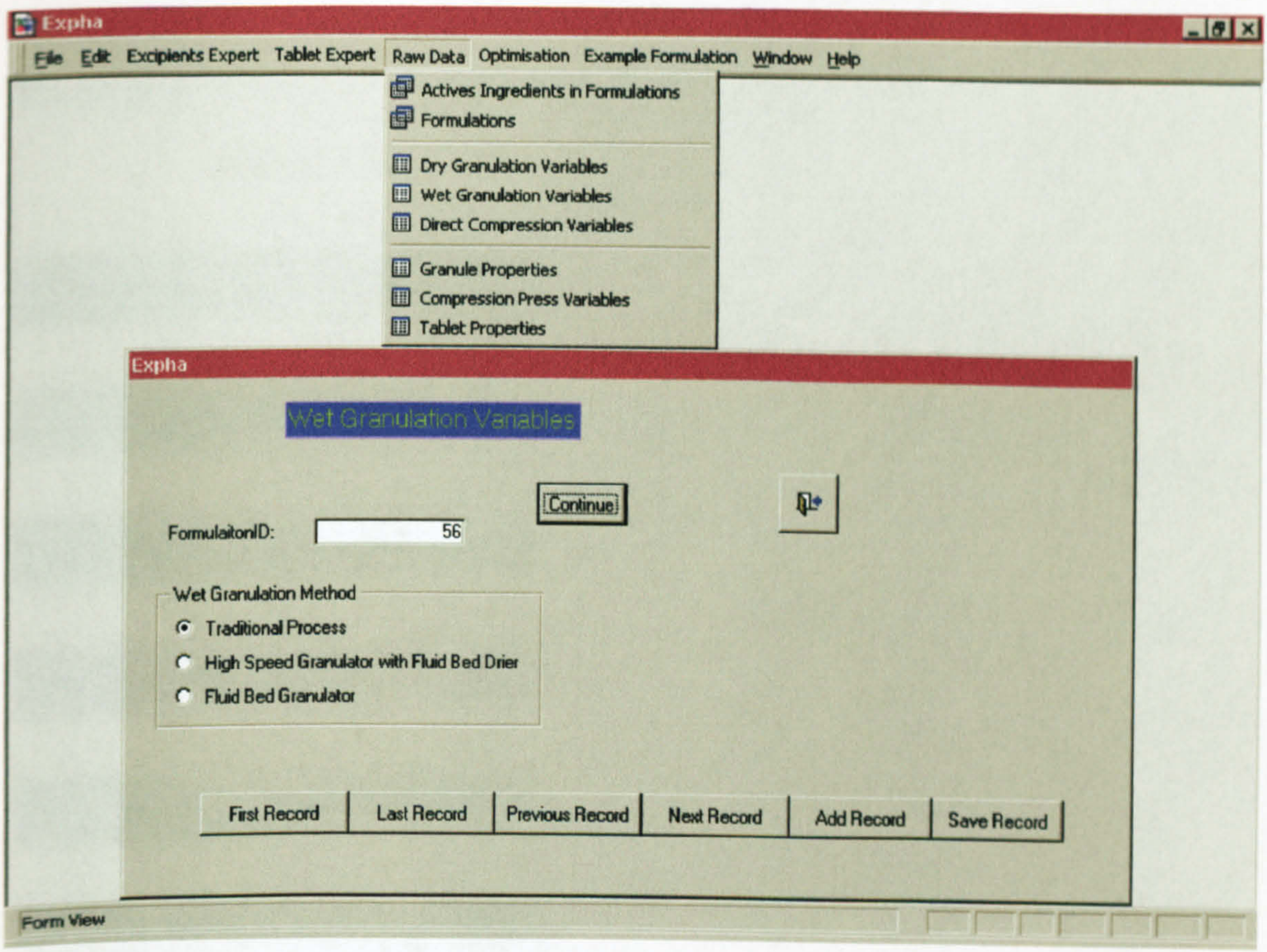


Figure 7.9: Wet granulation process variables form.

"Raw Data" menu includes the following tables:

"Active Ingredients in Formulations"

"Formulations"

"Dry Granulation Variables"

"Wet Granulation Variables"

"Direct Compression Variables"

"Granule Properties"

"Compression Press Variables"

"Tablet Properties"

A hypothetical example of how Patel's data (Patel, 1996) which was used in the tablets study, of Chapters 3 and 4, could be used by Expha will be explained in this section. From the "Raw Data" menu, choosing "Formulations" and "Compression Press Variables" will yield the two relevant sets of data for the independent variables. Specifically, from "Formulations" data set the two fields percentage of disintegrant and lubricant will be set, from "Compression Press Variables" data set the field of compaction force is set. "Tablet Properties" will supply data for the 8 responses like disintegration time etc.

To summarise, Figure 7.10 demonstrates the program flow:

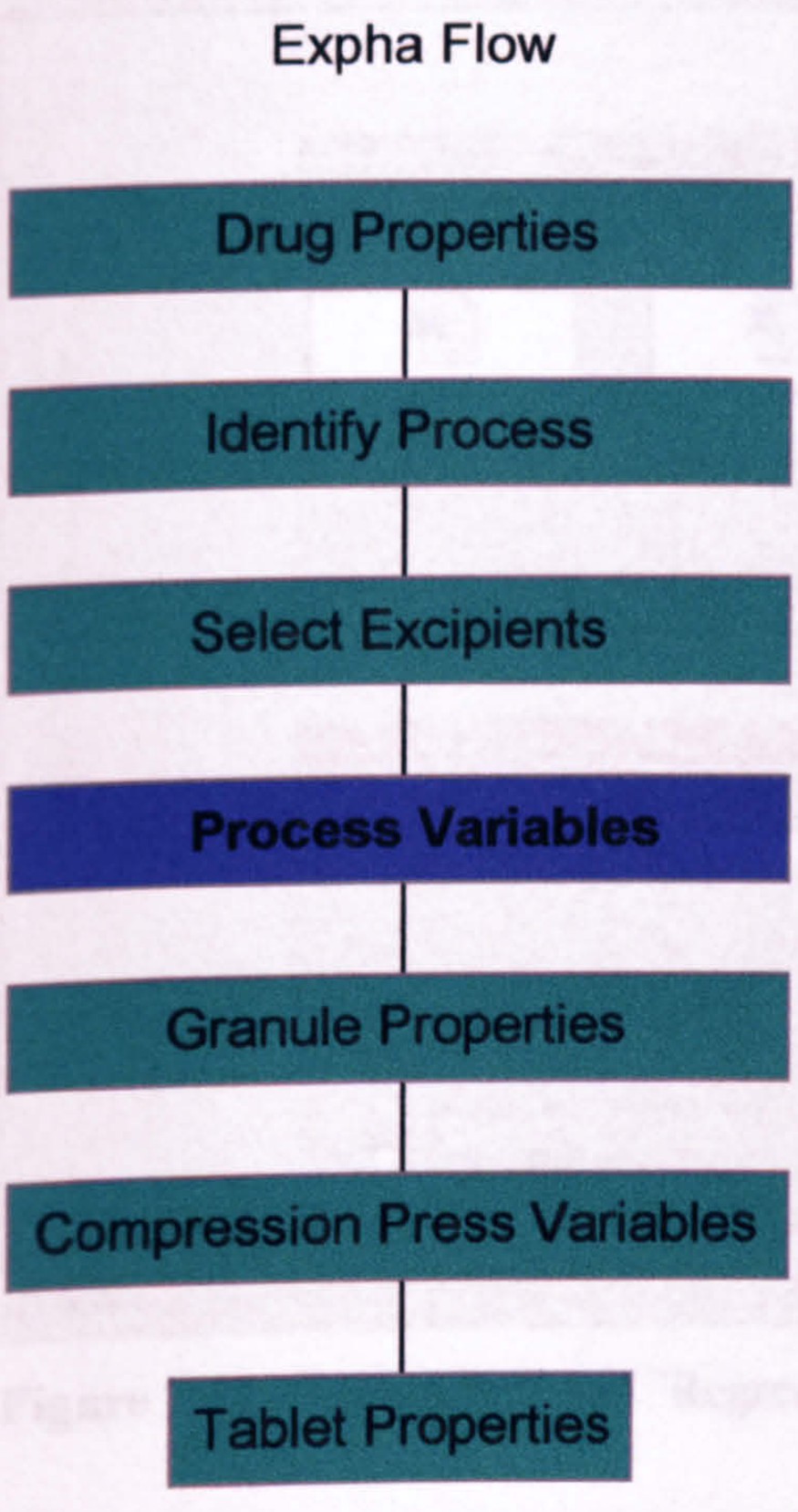


Figure 7.10: Expha flow. The boxes that are without text in bold (green boxes) relate to one form, whereas the one in bold (blue box) relates to multiple forms.

In Figure 7.10, each box represents one form except "Process Variables" box which represents several forms since the process variables change according to the process, and for each process there is one form. All these forms can be accessed through "Tablet Expert" menu. The flow is from top to bottom but the user can jump from one form to another not through the forms themselves but through the menu. This capability enables the user skip non-relevant forms, e.g. if granule properties were not measured the user can jump directly to "Compression Press Variables" form. After data have been collected it is now possible to build models for optimisation.

7.4.3 Data modelling and optimisation

After the data is collected it is possible to move to the stage of building models and optimisation. From the menu of "Optimisation" one can go to "Data Input" form, this yields

the top form in Figure 7.11.

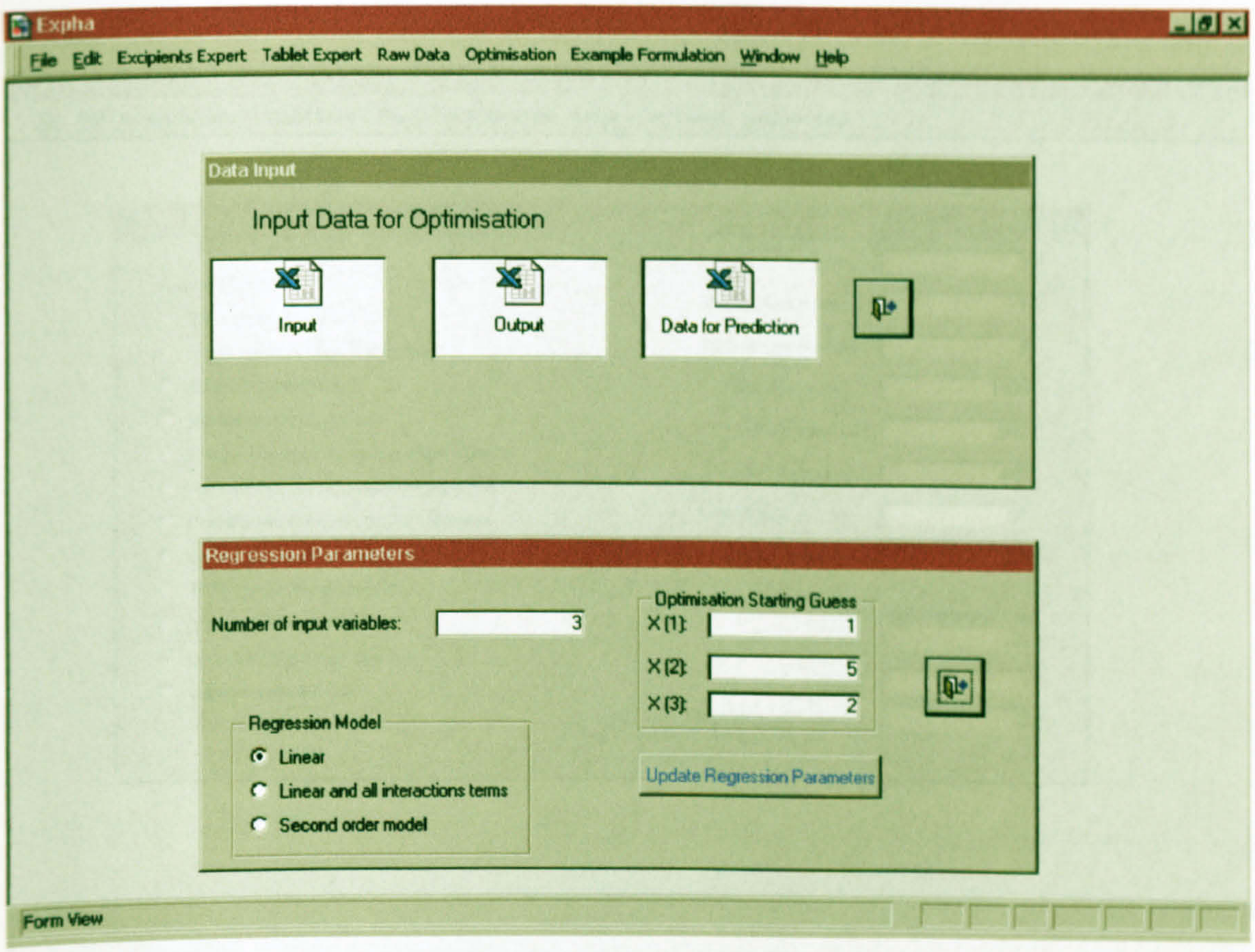


Figure 7.11: "Data Input" and "Regression Parameters" forms.

The user needs to copy data from the relevant fields in the "Raw Data" menu tables into the "Input" and "Output" embedded excel sheets. By opening the "Input" data file the user copies the independent variables data to this sheet. In the same manner the response data is copied to the "Output" data file. One can also put data of independent variables in "Data for Prediction" data file to get the program's predictions. From the "Optimisation" menu the user can choose "Regression Parameters" and the form that opens is at the bottom of Figure 7.11. On the left side of this form, the user fills in the parameters that build up the model, number of input variables and type of regression model. On the right side is the user's starting guess to the optimisation process. The number of input variables must correspond to the number of fields filled in for the starting guess, e.g. if the user enters just two input variables (in the relevant field, "Number of input variables") the optimisation starting guess must comply with this to have only two values.

Figure 7.12 shows the "ANN Parameters" form, which is invoked from "Optimisation" menu.

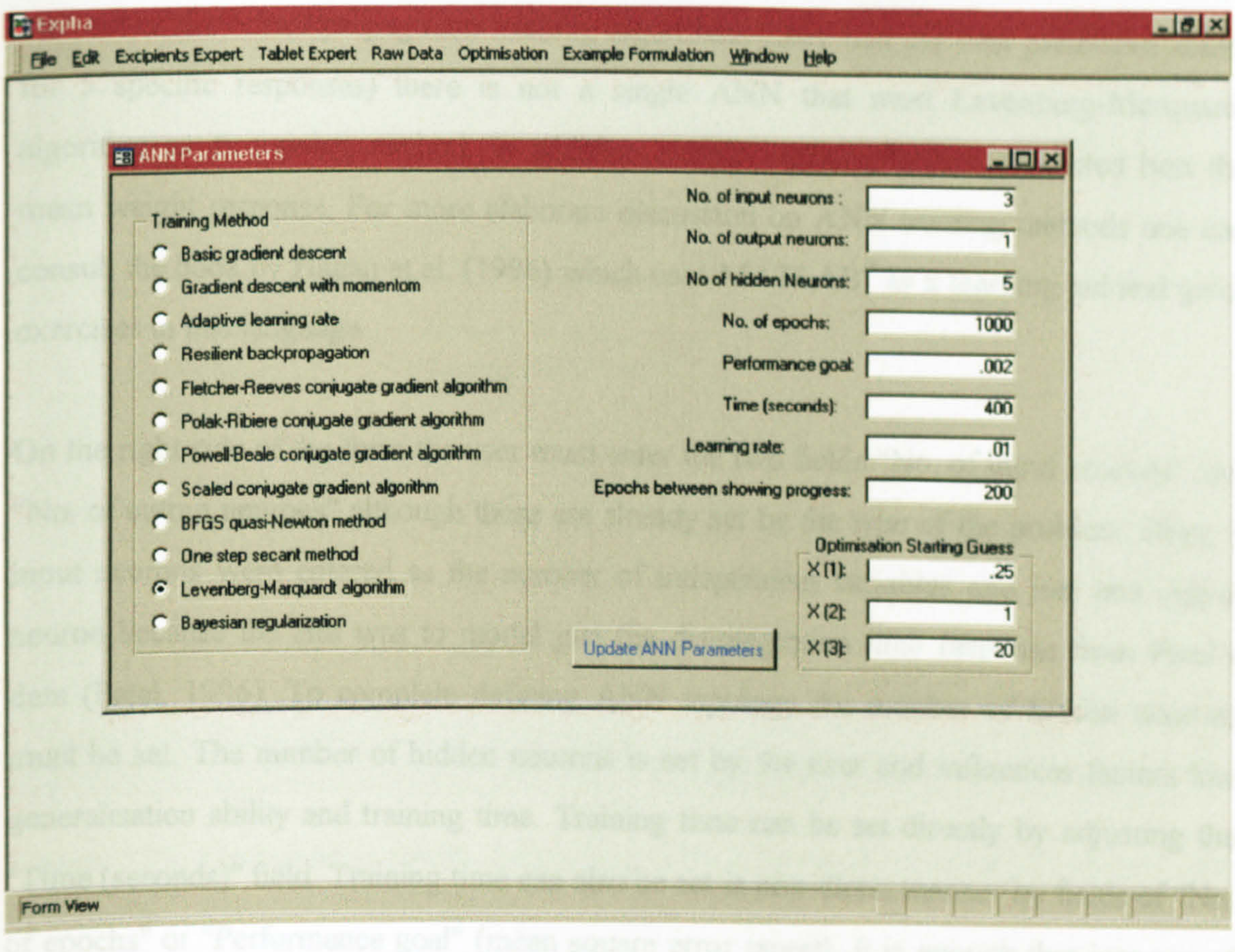


Figure 7.12: "ANN Parameters" form.

In the "ANN Parameters" form the user can choose from a variety of training methods. There is no one algorithm that is best suited for all purposes (Demuth & Beale, 1998). In terms of speed Levenberg-Marquardt is considered the fastest training algorithm for ANN of moderate size (up to a few hundreds weights). However, this statement is very general since the speed of convergence depends on a variety of factors, e.g. complexity of the problem, the number of data points in the training set, the number of weights and the performance goal (MSE goal). For example, if the performance goal is small, accurate training is required, so there is more chance the Levenberg-Marquardt algorithm will converge faster than other algorithms. The BFGS quasi-Newton is next in terms of convergence speed. This algorithm is generally faster than the conjugated gradient algorithms. Within this group of conjugate gradient algorithms the Powell-Beale algorithm is usually the fastest one (see Demuth & Beale, 1998 for comparison between the speed of learning algorithms). It is not always better to have fast convergence, it may be that ANN trained with a fast algorithm will not give as low a validation set error as the ones reached

by slow algorithms like simple backpropagation or simple variations of this algorithms. This study gives a good example to support this point; looking at Table 4.1, one can see that out of the 5 backpropagation-learning algorithms (they had the best predictive ability for 5 specific responses) there is not a single ANN that used Levenberg-Marquardt algorithm as its training method. In addition, simple backpropagation predicted best the mean weight response. For more elaborate discussion on ANN training methods one can consult the book by Hagan et al. (1996) which uses MATLAB® as a learning aid and gives exercises in this language.

On the right side of the form the user must enter the two fields "No. of input neurons" and "No. of output neurons" although these are already set by the type of the problem. Here, 3 input neurons were entered as the number of independent variables and just one output neuron because the aim was to model just the disintegration time response from Patel's data (Patel, 1996). To complete defining ANN topology the number of hidden neurons must be set. The number of hidden neurons is set by the user and influences factors like generalisation ability and training time. Training time can be set directly by adjusting the "Time (seconds)" field. Training time can also be set in non-direct manner by fields of "No. of epochs" or "Performance goal" (mean square error target). It is enough that just one of the latter three conditions is met to halt training. The next field, learning rate, is very important in training method of basic gradient descent backpropagation, but has no importance in backpropagation using adaptive learning rate. The reason for this is that in the latter method the learning rate is changing all the time through the learning process. The field labeled as "Epochs between showing progress" influences the amount of output that shows the progress of training in terms of the parameters—time in seconds, MSE and gradient. Like the form for "Regression Parameters" the user has to fill the starting guess for optimisation, located on the lower right side of the form. The "Update ANN Parameters" button causes optimisation results form to open. Figure 7.13 shows this form with a sub-form showing prediction and optimisation results generated after ANN training.

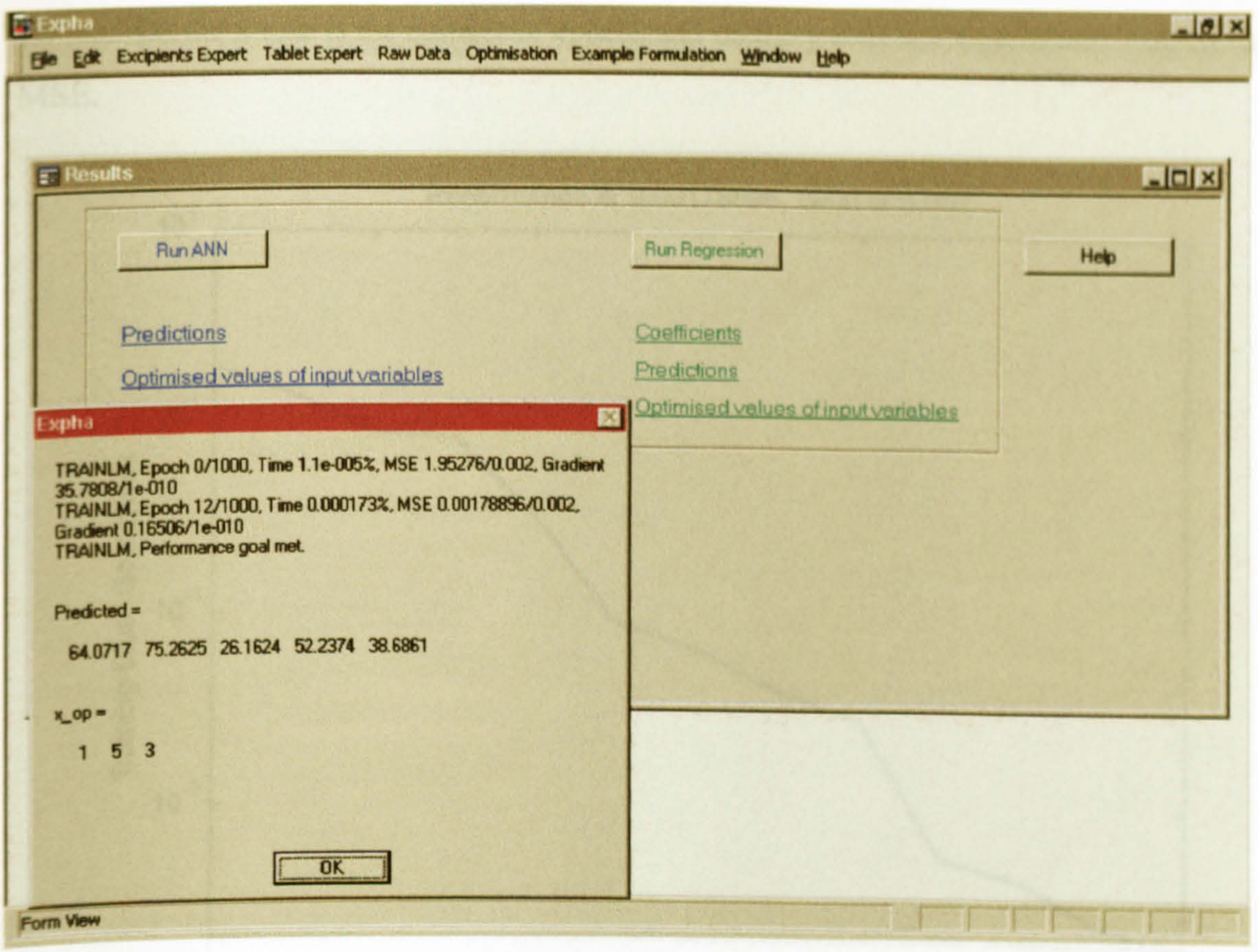


Figure 7.13: Optimisation results form along with a sub-form showing the results after running ANN and optimisation routines.

The "Run ANN" button runs the ANN training and optimisation routines. The two programs run sequentially. First training, then optimisation. "x_op" is output from the optimisation routine and the rest of the output is from the training phase. In a similar manner "Run Regression" causes the regression equation to be generated and afterwards the optimisation routine is invoked. The ANN in this example was trained with the ANN parameters set in the previous form. As mentioned before, the response data, which is disintegration time, were taken from the tablet study experiments, but the last 5 cases left out of the training and were used for prediction. The predictions by the ANN is shown in the sub-form ("Predicted =") along with "x_op" which show the optimised independent variables values. The optimised independent variables are 1% lubricant, 5% disintegrant and compaction force of 3 kN (Figure 7.13). The permitted range of values (constraints) for the independent variables in this optimisation routine were: 0.25-2%, 1-5%, 3-20 kN for lubricant, disintegrant and compaction force respectively. Pressing "OK" on the subform does not mean that this important data is lost since below the "Run ANN" button there are two hyperlinks connected to files which store the predicted and optimised values. After pressing "Run ANN" there is also a graph (Figure 7.14) that shows the progress of training

on-line. On the x-axis is the number of epochs and on the y-axis is a logarithmic scale of the MSE.

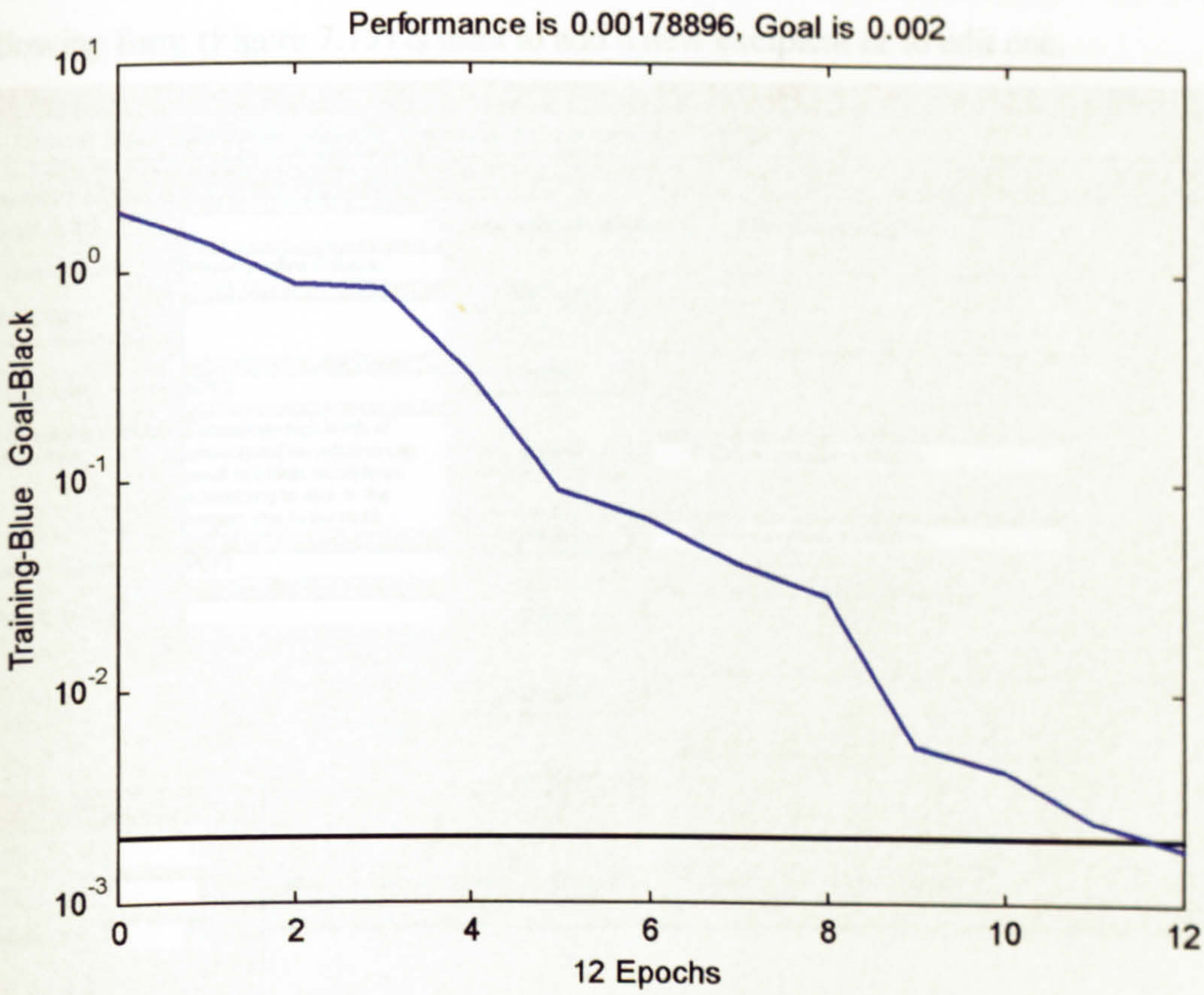


Figure 7.14: MSE (mean square error) as a function of number of epochs plot. The horizontal line in black is the target MSE and in blue is the training progress.

The plot in Figure 7.14 was generated using ANN trained with the Levenberg-Marquardt algorithm, with parameters as defined in "ANN Parameters" form presented in Figure 7.12. The plot corresponds to the optimisation results sub-form that was shown in Figure 7.13. As can be seen, the training stopped after 12 epochs since it achieved the performance goal. After the training phase of ANN it can be saved to be used in decision-making in Expha, e.g. the user can train ANN to predict the stability of a drug in a granulation process and this ANN could be used in a later stage.

7.4.4 Database maintenance using "Excipients Expert" menu

This section explains how to maintain the excipients database.

The following form (Figure 7.15) is used to add a new excipient or to edit one.

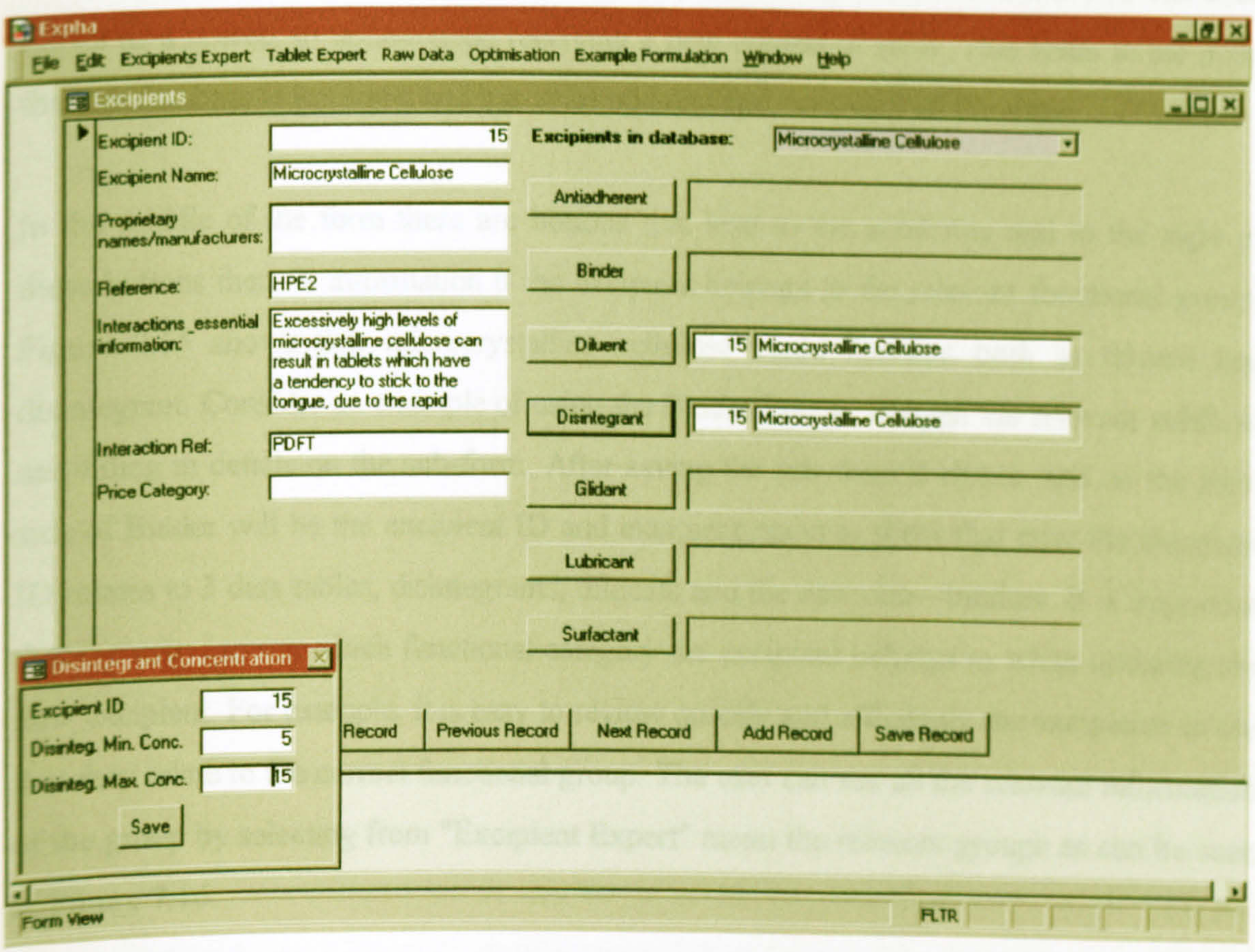


Figure 7.15: "New/Edit Excipients" form.

This form can be viewed by choosing "Excipients Expert" menu then selecting "New/Edit Excipients". By using the blue combo box on the top right of the form microcrystalline cellulose was selected. The left side is used for filling data. "Excipient ID" (field) is a unique number, which connects the excipient identity number to other tables. Hence, the excipients database is a relational one. "Reference" field is the source for data used in the subform(s) (the *s* in parhentesis since some excipients belong just to one functional category or one subform). Figure 7.15 shows an example of how to use the functional category buttons; the "Disintegrant" button was selected in order to open the "Disintegrant Concentration" subform and fill in the minimum and maximum disintegrant concentration. The data for these concentrations was taken from Handbook of Pharmaceutical Excipients (HPE) (Wade & Weller, 1994). Hence the relevant field has the value "HPE2" (2 standing for second edition). There is also a second field of reference that relates to the source of

interactions and essential information. In this case it is from Pharmaceutical Dosage Forms: Tablets (PDFT) (Lieberman et al., 1989). It is important that the user fills in details in the same manner. In this case, the user must provide adequate reference to allow for efficient data searching and updating. For example, a user could request all the excipients that used HPE2 as a source, to update upon receiving a new edition of HPE. This leads to the point that the database is not static and has to be updated and reviewed all the time.

In the middle of the form there are buttons that lead to the subforms and to the right of these buttons there is information if the excipient belongs to the relevant functional group. Figure 7.15 shows that microcrystalline cellulose could function both as diluent and disintegrant. Consider an example of using the Binder button, to open the relevant subform and filling in details on the sub-form. After saving the sub-form it closes, and on the right side of Binder will be the excipient ID and excipient name to show that now the excipient ID relates to 3 data tables, disintegrants, diluents and the new one—binders. It is important that the user can see which functional category the excipient belongs to when updating the new excipient. For example, it is easy to review quickly and efficiently the excipients to see that they relate to the correct functional group. The user can see all the relevant information of the group by selecting from "Excipient Expert" menu the relevant groups as can be seen in Figure 7.16.

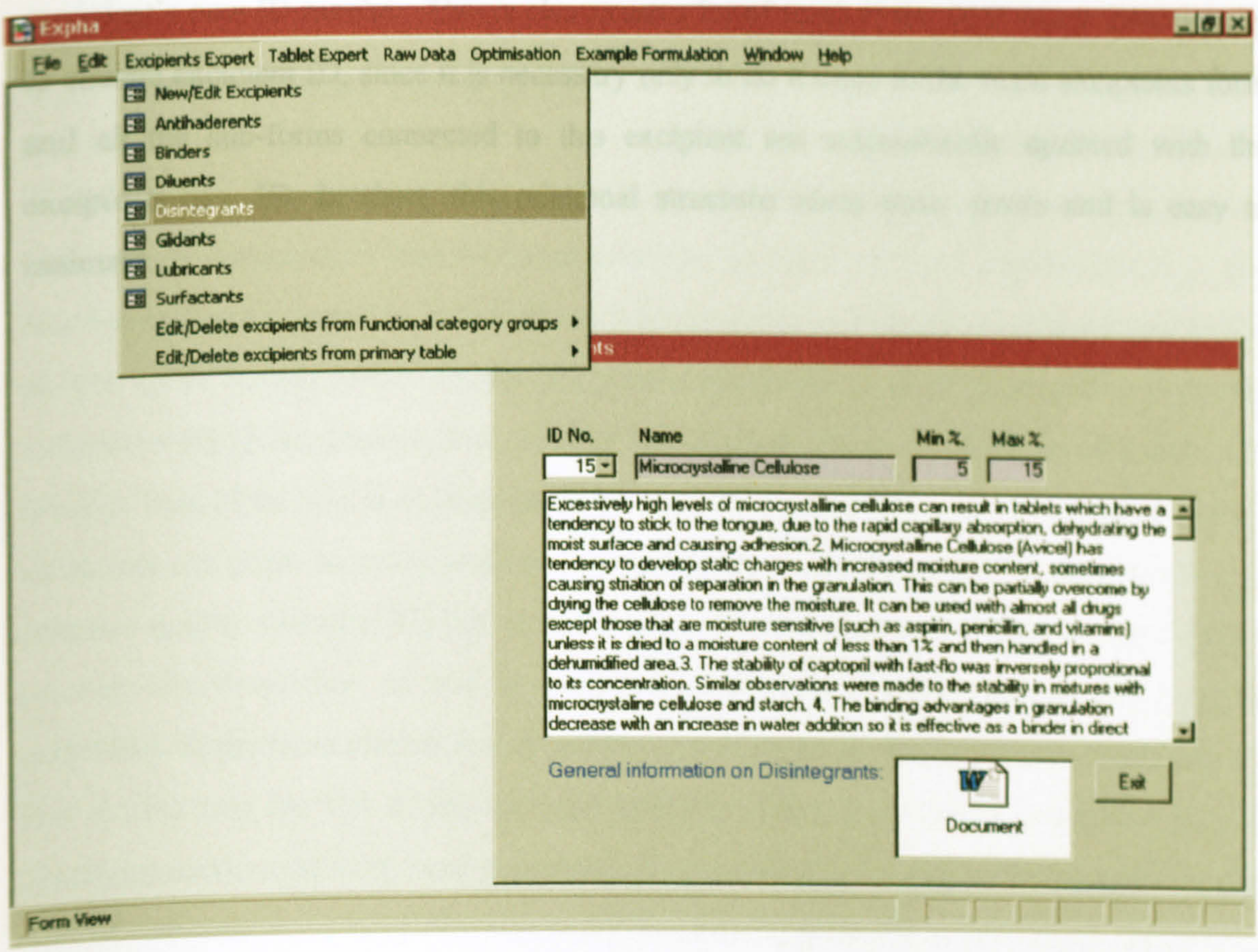


Figure 7.16: "Disintegrants" information form.

"Disintegrants" information form is only for viewing. The combo box on the top left side allows the user to see a list of all disintegrants. The minimum and maximum concentration as well as interactions and important information relevant for the specific excipient are displayed in the big text box. There is also an embedded MSWord® document, which gives general information on the relevant functional category.

The "Excipient Expert" menu is a revealing demonstration of the power of a properly designed relational database. If the user tries to delete unintentionally the wrong excipients from the primary table using the menu "Edit/Delete excipients from primary table" Expha does not allow this deletion because it relates to other tables as well. So the user first has to delete the connection of the excipient to other tables using the menu "Edit/Delete excipients from functional category group". After breaking links it is then possible to delete the excipient from the primary table. This behaviour of Expha prevents chunks of data that are not connected to any excipient, from being left in the database. If a user changes an excipient ID by mistake, the data which relates to this excipient is not lost because all the tables which the excipient is connected to also update their reference according to the

excipient's new ID number. This is also saves a lot of work if the user wants intentionally to change excipient ID, since it is necessary only to do it once in the main excipients form and all the sub-forms connected to this excipient are automatically updated with the excipients new ID. In short, this relational structure saves time, errors and is easy to maintain.

7.5 Testing and field trial

There is a well-known phrase that a person in a good health is one whose condition is yet to be correctly diagnosed. Using the same principle, software without bugs/problems is one that has not been adequately tested. There is surely more truth in this last postulate than the phrase about human health. In the pharmaceutical industry, each factory has a quality assurance (QA) department that verifies the pharmaceutical product is of acceptable quality. Part of its role is to inspect the manufacturing process and to identify sensitive areas that are prone to cause problems in the product's quality. The QA department also contains quality control (QC) that is responsible for all the analytical tests of the finished product like dissolution as well as analytical tests of in-process material like granule properties. In the same manner that QA of the pharmaceutical world is a very broad subject that is also true for QA in the software industry. Although it might seem that QA in pharmaceutical world is of more importance it is not always the case since computers also control real-time equipment that if it is not functioning properly would cause the loss of human life, e.g. hospitals real-time systems. There was no attempt to perform professional QA work on Expha expert system since there is no in-house expertise and the tools a software engineer specialised in this field would have were not available. Nevertheless, there was an attempt to employ part of the QA philosophy of the pharmaceutical industry to trace critical points through the manufacturing process (of the software in this case). In order not to end up with a product that has so many bugs it is not convenient to work with.

In general, everything that was programmed to do some sort of operation was tested. For example, the advice button on the "Select Excipients" form was checked to ensure it gave the advice it was supposed to give and not advice that is not updated in accord with a change in data. Another example is that the algorithm for manufacturing process was tested. For the latter purpose different drug properties settings were used to give assurance that Expha could guide the user for all manufacturing process possibilities.

There was extensive testing of Expha calculations, each calculation result being tested by comparing the Expha calculations with manual ones. Parts of the calculations are in critical areas; checking that the calculations of Drug Properties form are done properly, e.g. stability calculations are important part of Expha. After entering K and temperature values

the computer calculates the percent of drug remaining after 5 hours at 50° Celsius. From Figure 7.3 for k of 5×10^{-4} at a temperature of 100° Expha calculated that 99.975% remains in those conditions. The manual calculation explained earlier yields the same result as the computer.

This section will present tests that were conducted to verify appropriate functioning of model building and optimisation. Appropriate functioning means trying to trace inappropriate interactions between the underlying computer program modules within Expha. The emphasis here is on the correct functioning of the various modules on selection of appropriate options in the dialog boxes *and not on the numerical values that maybe produced for our test problem.*

A description of the experiment settings follows. The aim of the regression/ANN modelling part was to learn the equation $y = x^2 - 3$. For that purpose Expha was given x values from zero to twenty as input data and output data was the corresponding y values generated from the latter equation. To test the success of learning, x values of 1, 2, 3, 4, 5 were put as the set for prediction. The starting guess for the optimisation process was 100. The minimum/maximum point is the goal for the optimisation process. It is calculated from the derivative of the equation ($= 2x$) and the second derivative ($= 2$) indicates if it is a minimum or maximum point. Regarding the equation discussed, the optimised x value yields a minimum (since 2 is bigger than 0) point of 0. The chosen regression model was second order with interaction terms. The topology for ANN was one input neuron and one output neuron (it stems from the nature of the problem constraints) with 5 neurons in the hidden layer. The number of neurons was chosen as five to enable enough flexibility but to avoid overfitting due to too many weights. All twelve training methods available on Expha were tested. The number of epochs was set to 1000 and the second criteria to stop training was MSE (mean squared error) of 0.002.

Figure 7.17 presents Expha output after the regression module was run. "a_coefficients" are the regression coefficients of the regression equation, which is presented under the header "Equation =", starting with a(0) and in ascending order. "Predicted" are the predicted responses for independent variable x values. "x_op" are the optimised values of the input variables which minimise the dependent variable. The latter two terms "Predicted" and "x_op" are also relevant to final output of ANN results screens presented in figures

7.19, 7.21, 7.23, 7.25, 7.26, 7.27. The latter screens relate to the following training algorithms respectively: gradient descent with a maximum of 100 epochs (all other ANN training algorithms used 1000 epochs), gradient descent with a maximum of 1000 epochs, Levenberg-Marquardt algorithm and Bayesian regularization (Figures 7.25 - 7.27). The difference in the latter three figures is that the values for prediction in Figure 7.25 are like all other ANN that were trained. Whereas in Figure 7.26 the values for prediction are between 0.5 to 19.5 in increments of 1 (0.5, 1.5, ...19.5) and in Figure 7.27 the set for prediction is 60, 70, 80, 90, 100. Figures 7.18, 7.20 and 7.22 show learning curve plots for gradient descent with a maximum of 100 epochs, gradient descent with a maximum of 1000 epochs and Levenberg-Marquardt algorithm respectively. On these plots the y-axis is the mean squared error (MSE) and the x-axis is the number of epochs. The black line is the target performance representing MSE of 0.002. Figure 7.24 shows learning curve of Bayesian regularization training method. As in the other plots the x-axis represents the number of epochs (in all the three plots of this figure). The y-axis of the three plots from the upper to the bottom one stand for sum squared error (SSE), sum squared weights (SSW) and effective number of parameters. Table 7.1 summarises the results that were presented earlier in this section. Table 7.2 shows the predictions of additional ANN training methods. In both tables on the left column are the independent x values and on the second column are the target values according to the equation $y = x^2 - 3$. The last row in the tables represents the number of epochs of each ANN setting.

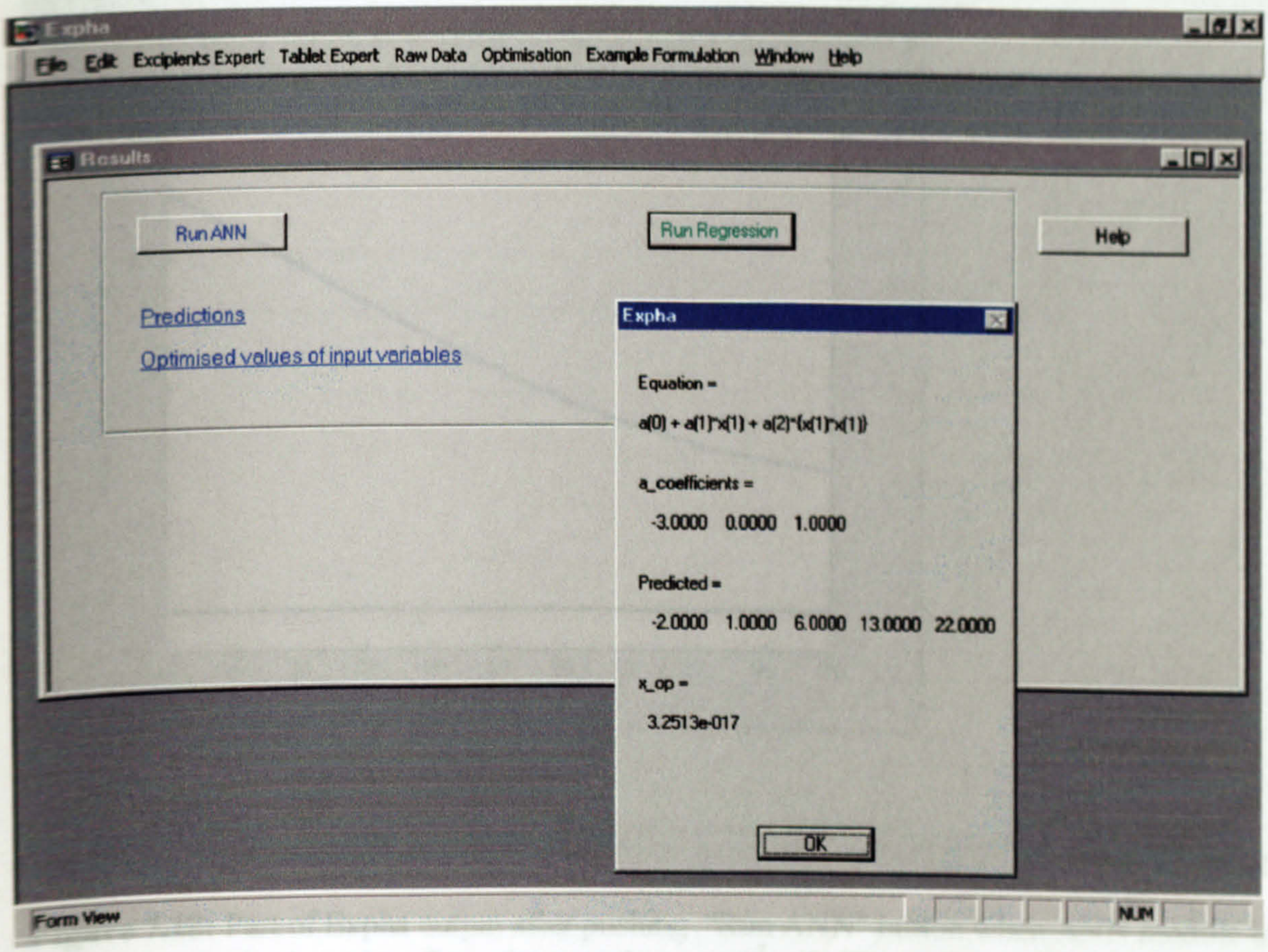


Figure 7.17: Pushing "Run Regression" button yields this output when regression module in Expha was trained on data for the equation $y = x^2 - 3$. The coefficients of the regression equation ("a_coefficients") and the predicted responses ("Predicted") are shown together with the optimal value of the input variable ("x_op").

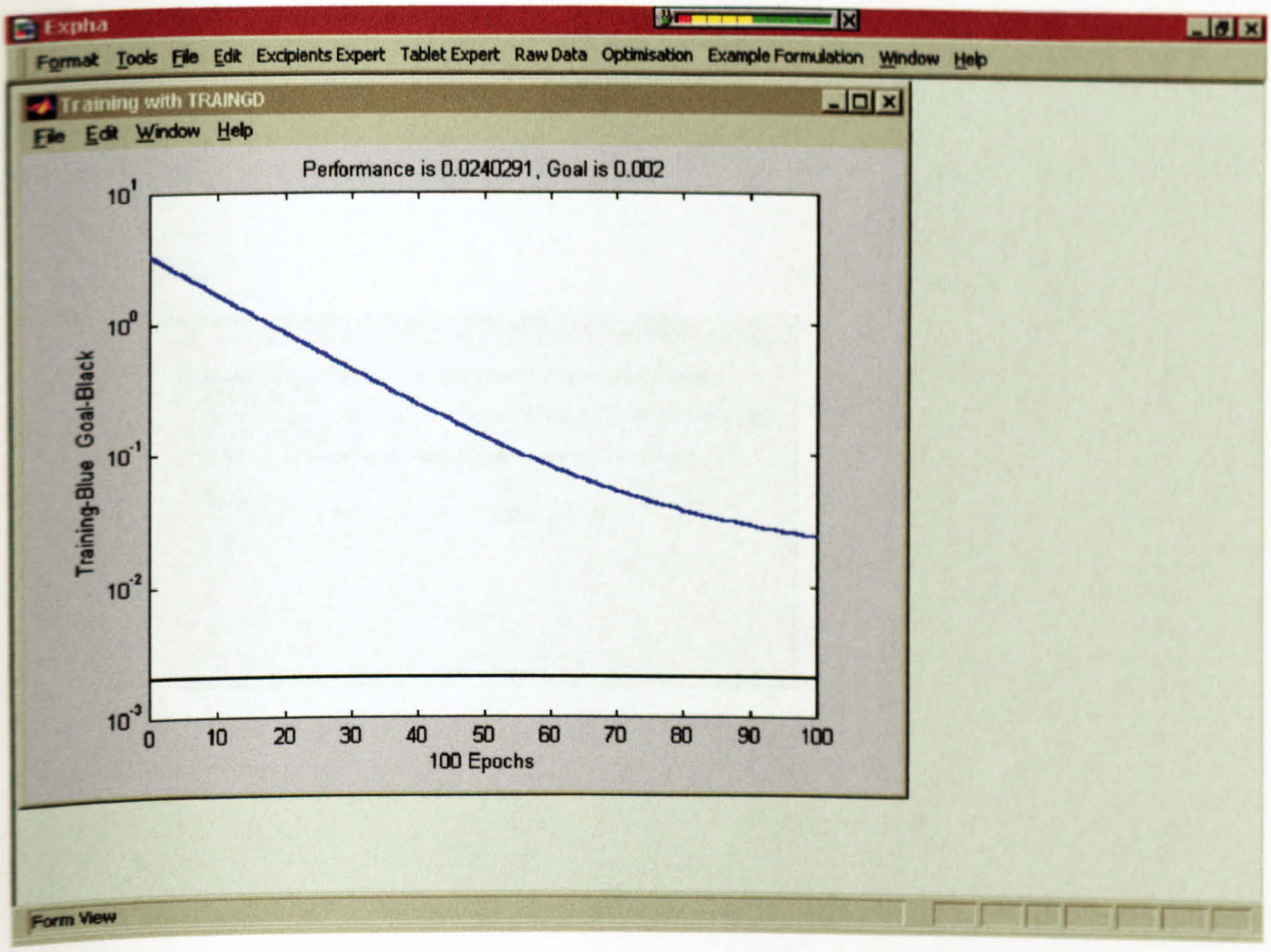


Figure 7.18: Part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with backpropagation using gradient descent with a maximum of 100 epochs. On the y-axis is the mean squared error and on the x-axis is the number of epochs. The black line is the target performance representing MSE of 0.002.

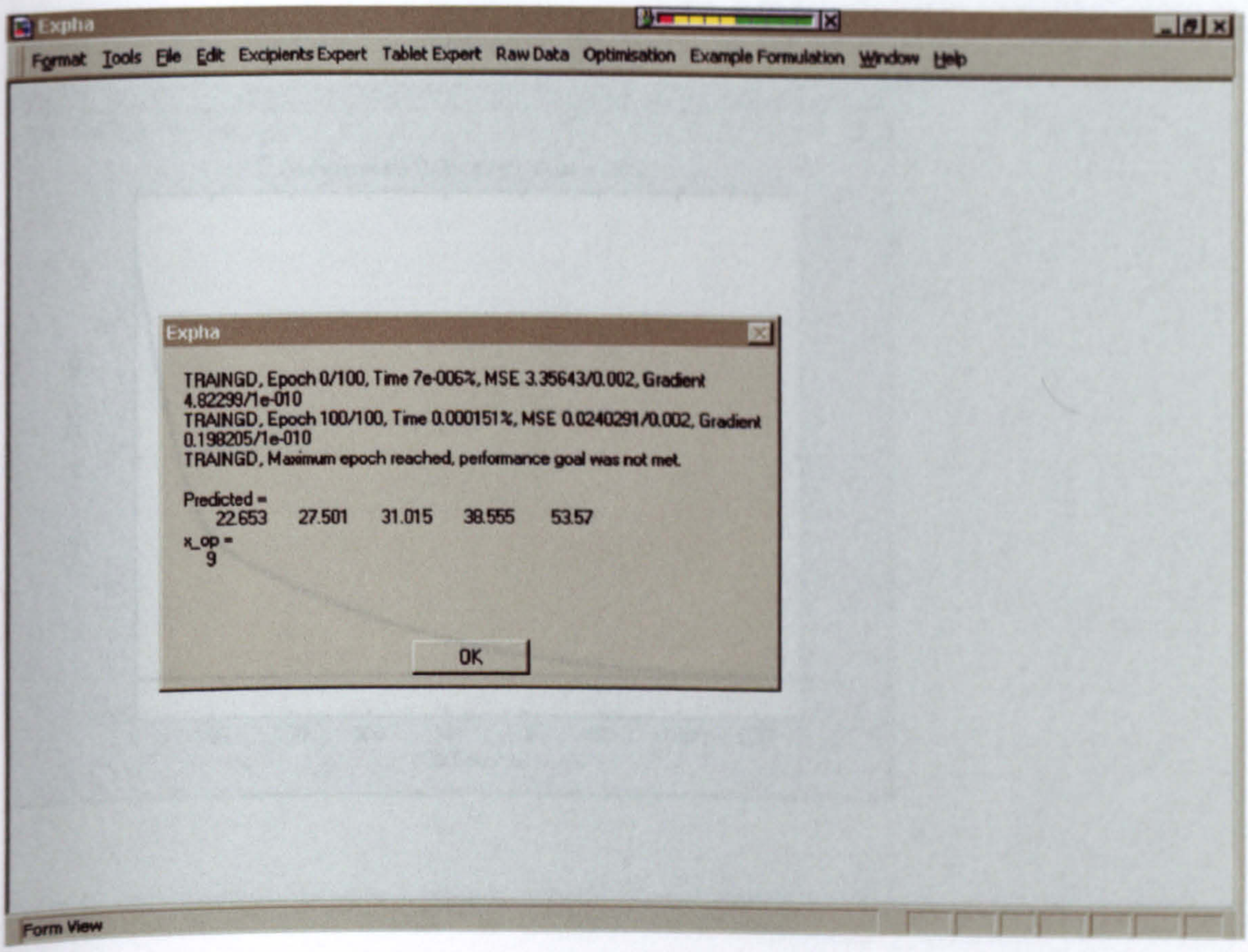


Figure 7.19: Final part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with backpropagation using gradient descent with a maximum of 100 epochs. "Predicted" are the predicted responses for independent variable x values of 1, 2, 3, 4, 5. "x_op" is the optimised value of the input variable which minimises the dependent variable.

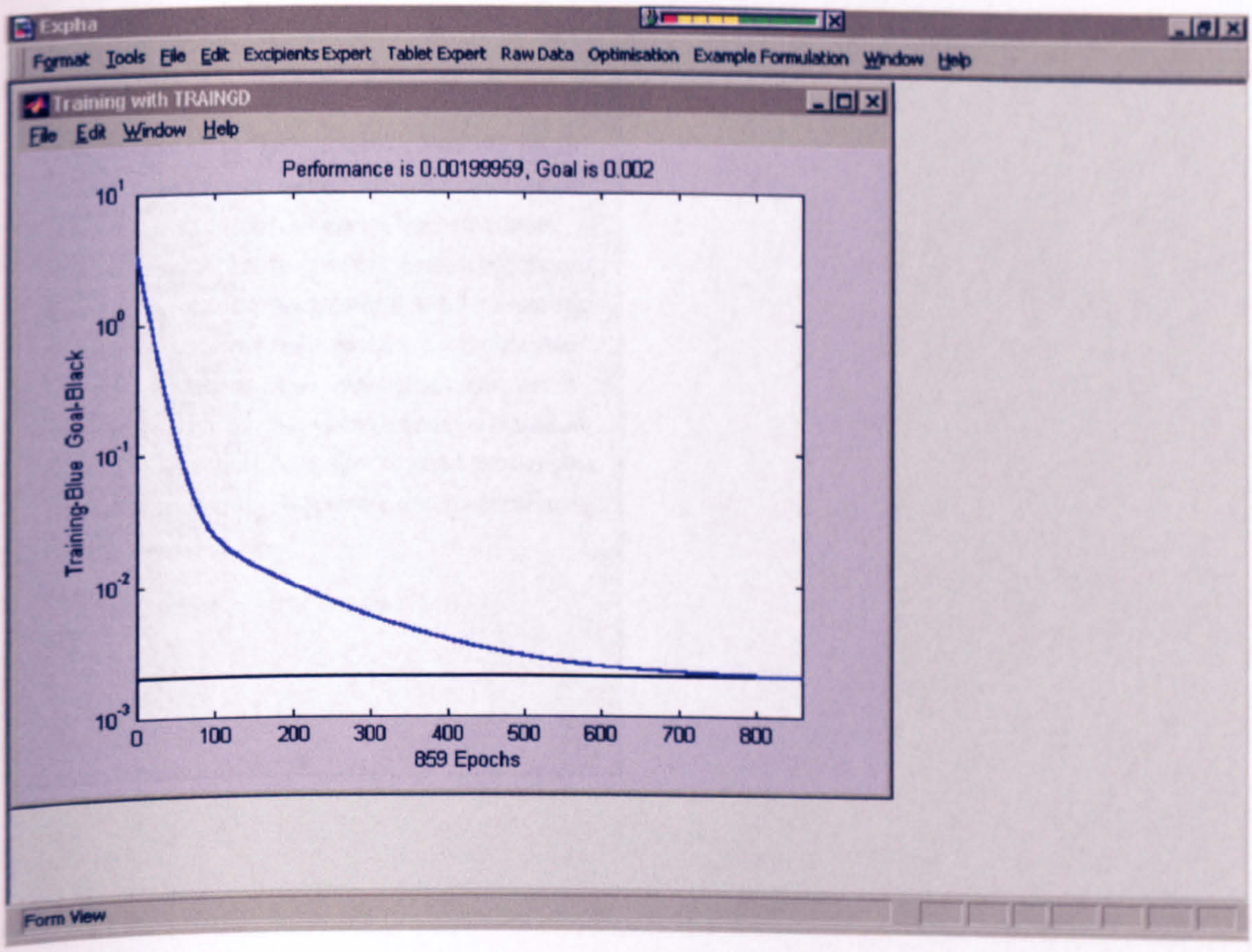


Figure 7.20: Part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with backpropagation using gradient descent with a maximum of 1000 epochs. On the y-axis is the mean squared error and on the x-axis is the number of epochs. The black line is the target performance representing MSE of 0.002.

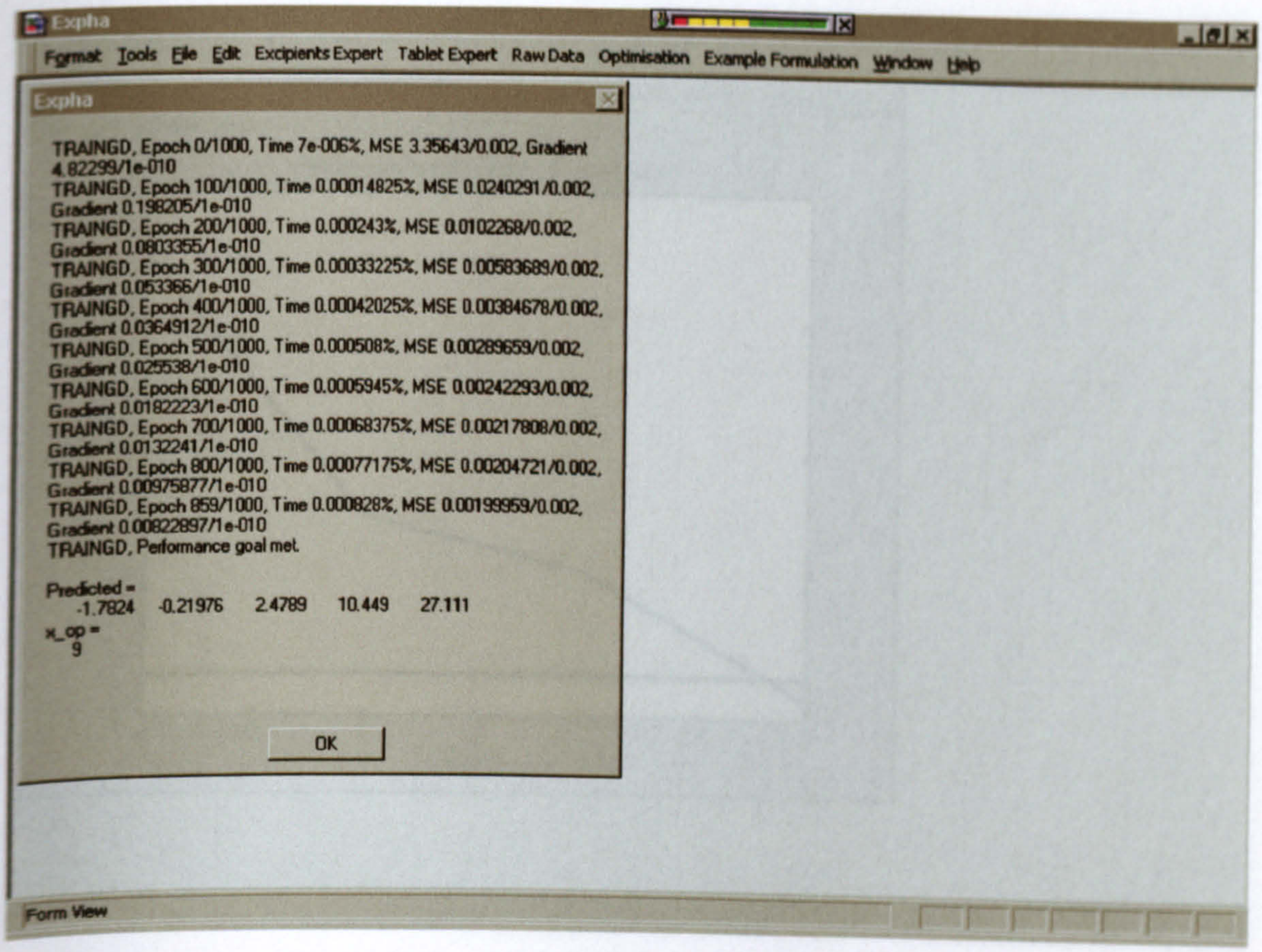


Figure 7.21: Final part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with backpropagation using gradient descent with a maximum of 1000 epochs. "Predicted" are the predicted responses for independent variable *x* values of 1, 2, 3, 4, 5. "x_op" is the optimised value of the input variable which minimise the dependent variable.

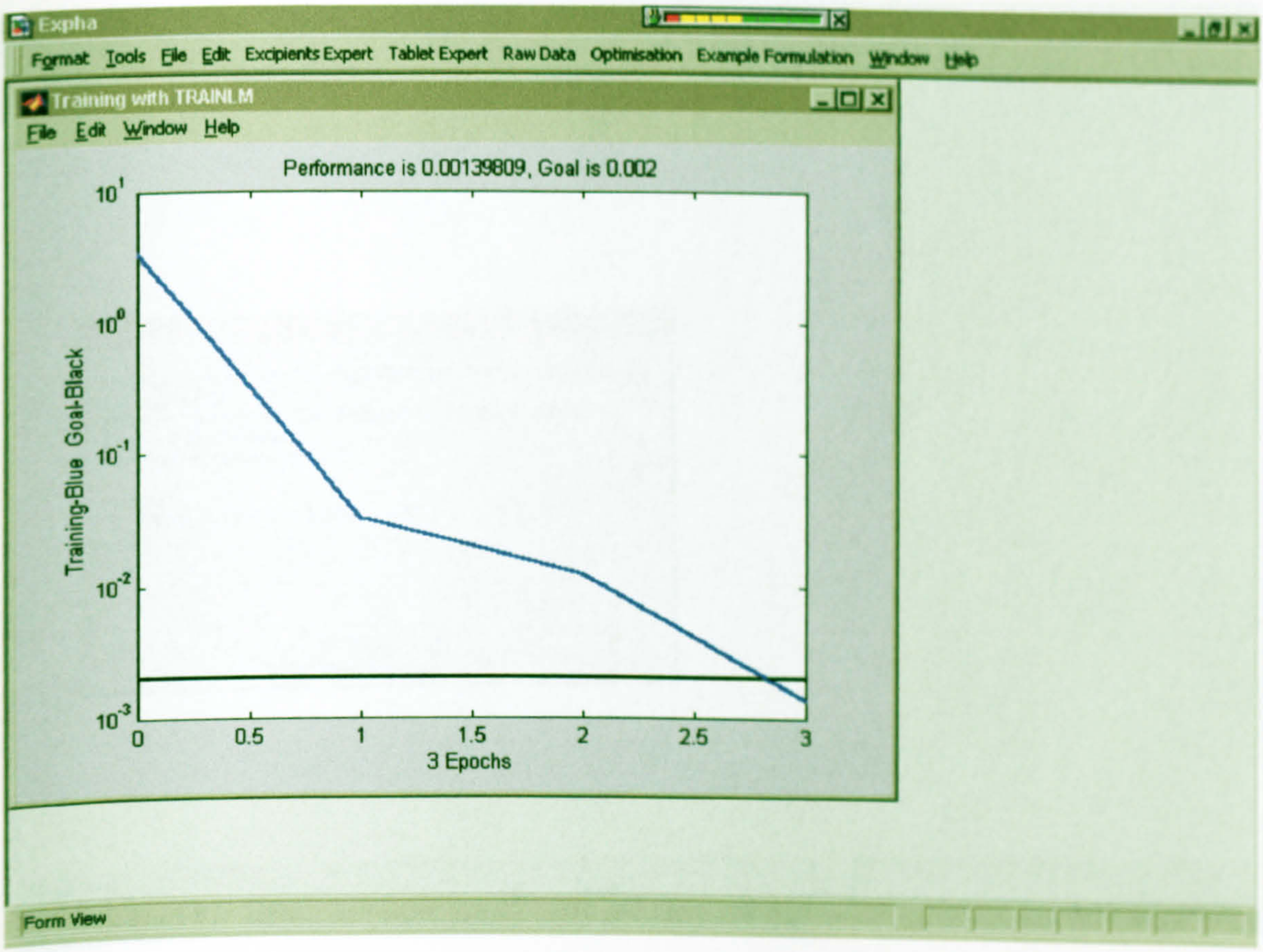


Figure 7.22: Part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with Levenberg-Marquardt algorithm with a maximum of 1000 epochs. On the y-axis is the mean squared error and on the x-axis is the number of epochs. The black line is the target performance representing MSE of 0.002.

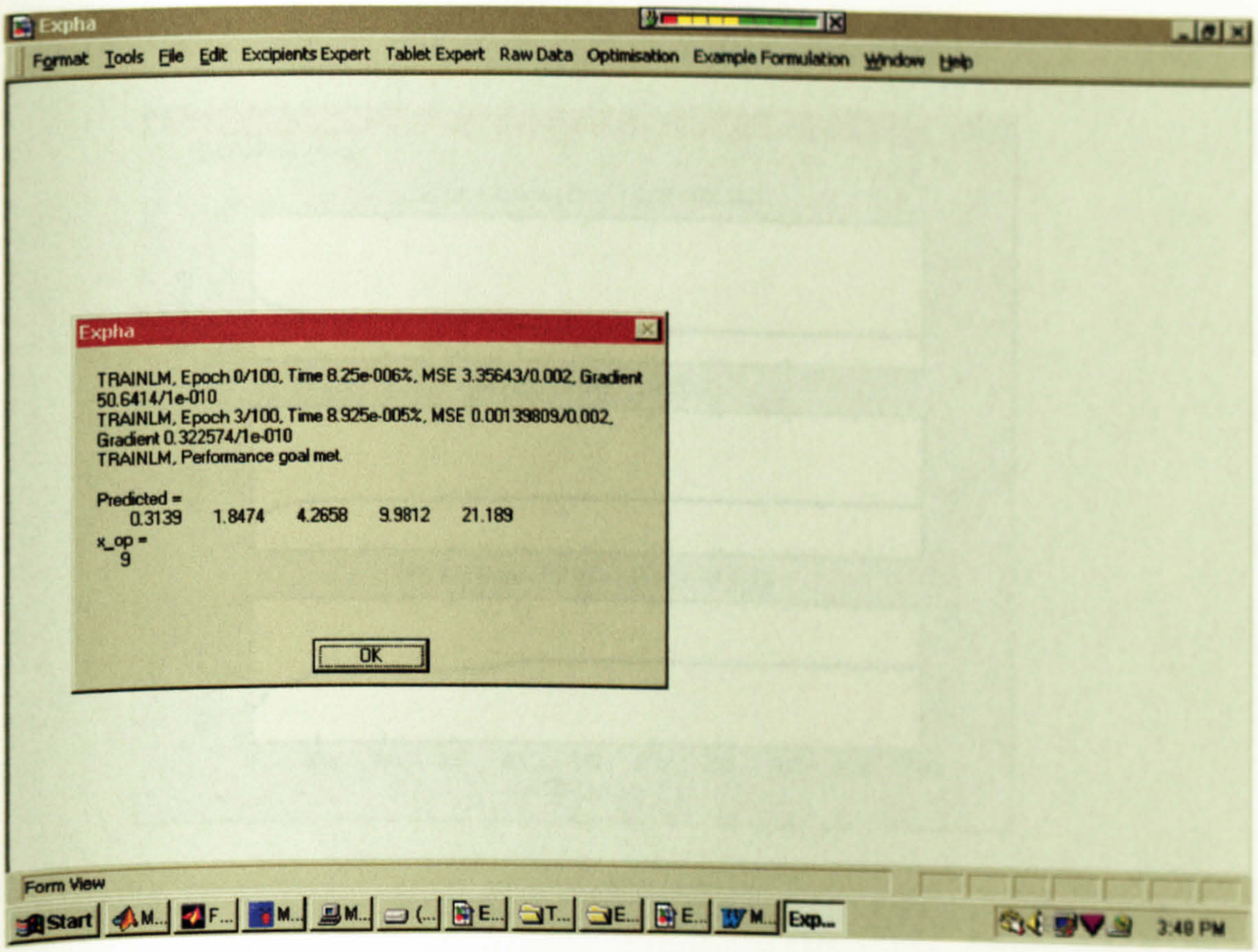


Figure 7.23: Final part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with Levenberg-Marquardt algorithm with a maximum of 1000 epochs. "Predicted" are the predicted responses for independent variable x values of 1, 2, 3, 4, 5. "x_op" is the optimised value of the input variable which minimise the dependent variable.

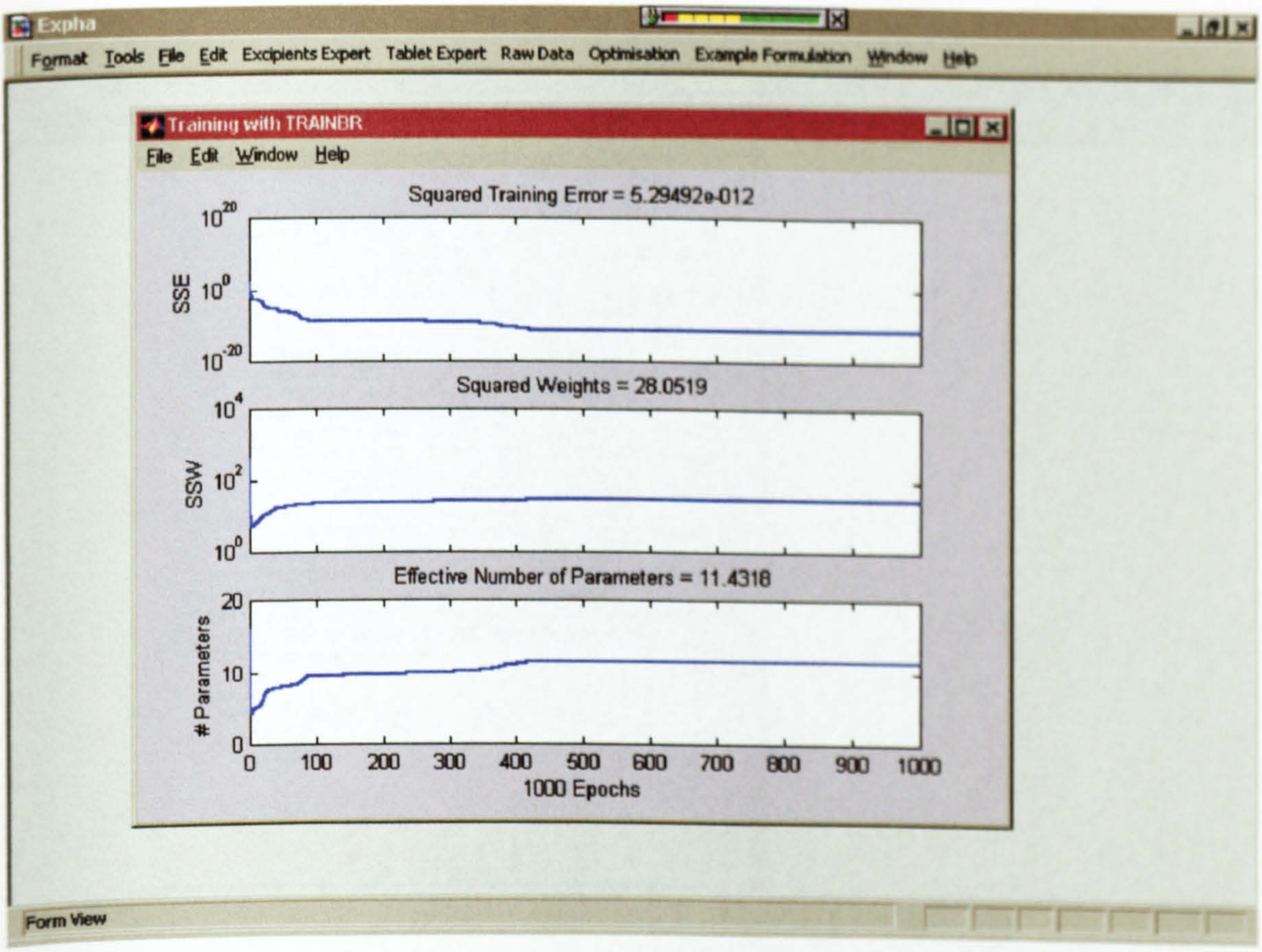


Figure 7.24: Part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with Bayesian regularization with a maximum of 1000 epochs.

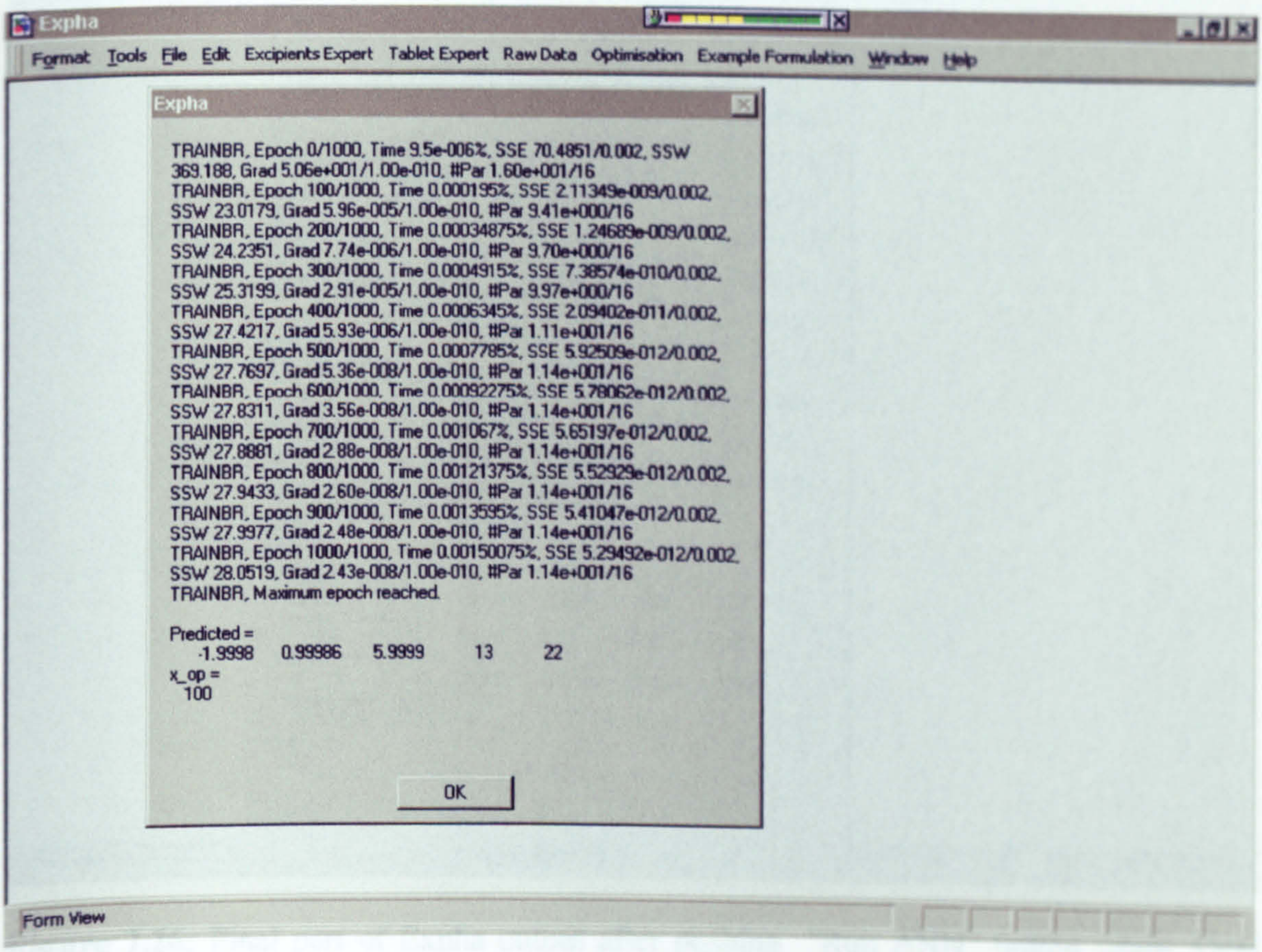


Figure 7.25: Final part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with Bayesian regularization with a maximum of 1000 epochs. "Predicted" are the predicted responses for independent variable x values of 1, 2, 3, 4, 5. "x_op" is the optimised value of the input variable which minimise the dependent variable.

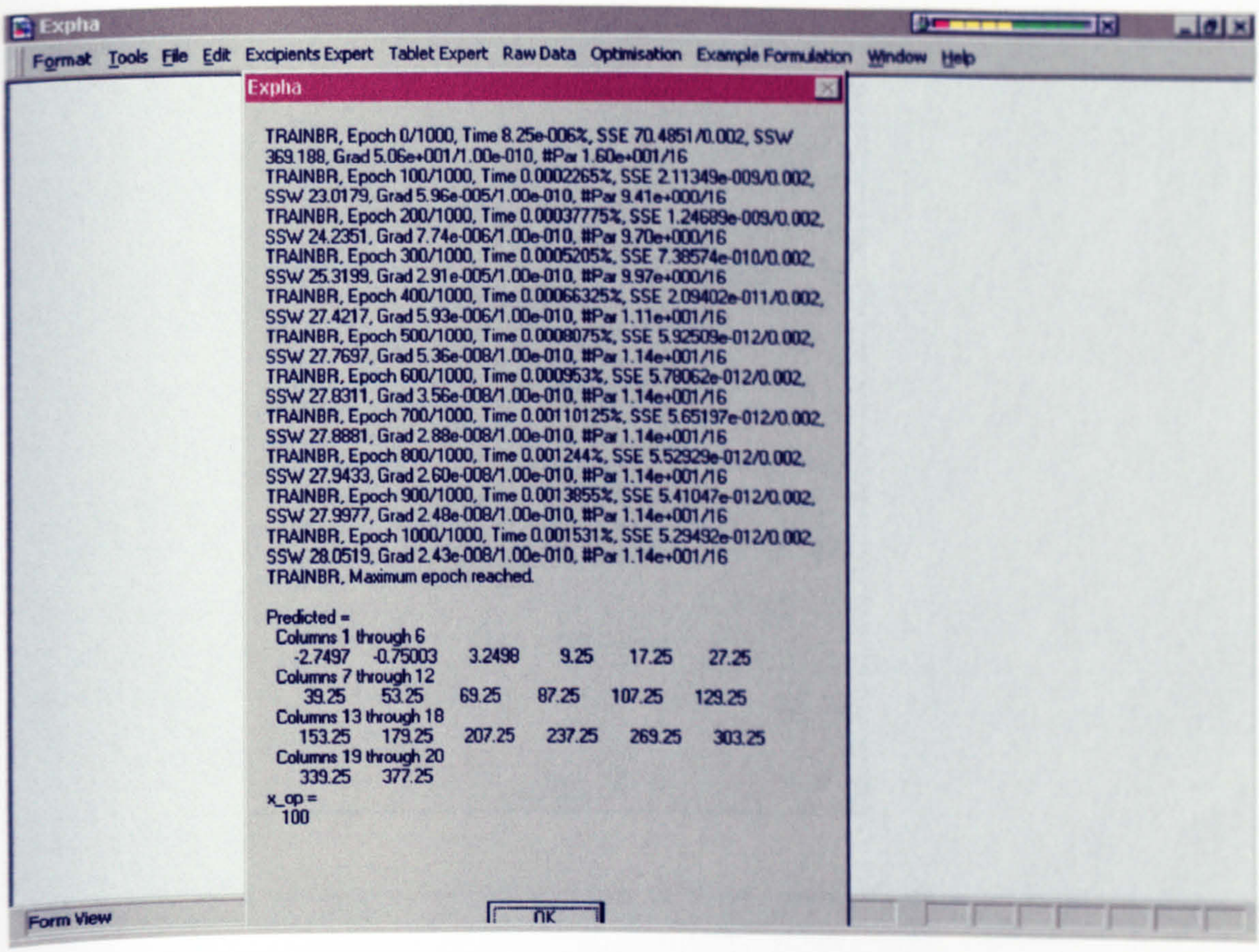


Figure 7.26: Final part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with Bayesian regularization with a maximum of 1000 epochs. "Predicted" are the predicted responses for independent variable x values between 0.5 to 19.5 in increments of 1 (0.5, 1.5, …19.5). "x_op" is the optimised value of the input variable which minimise the dependent variable.

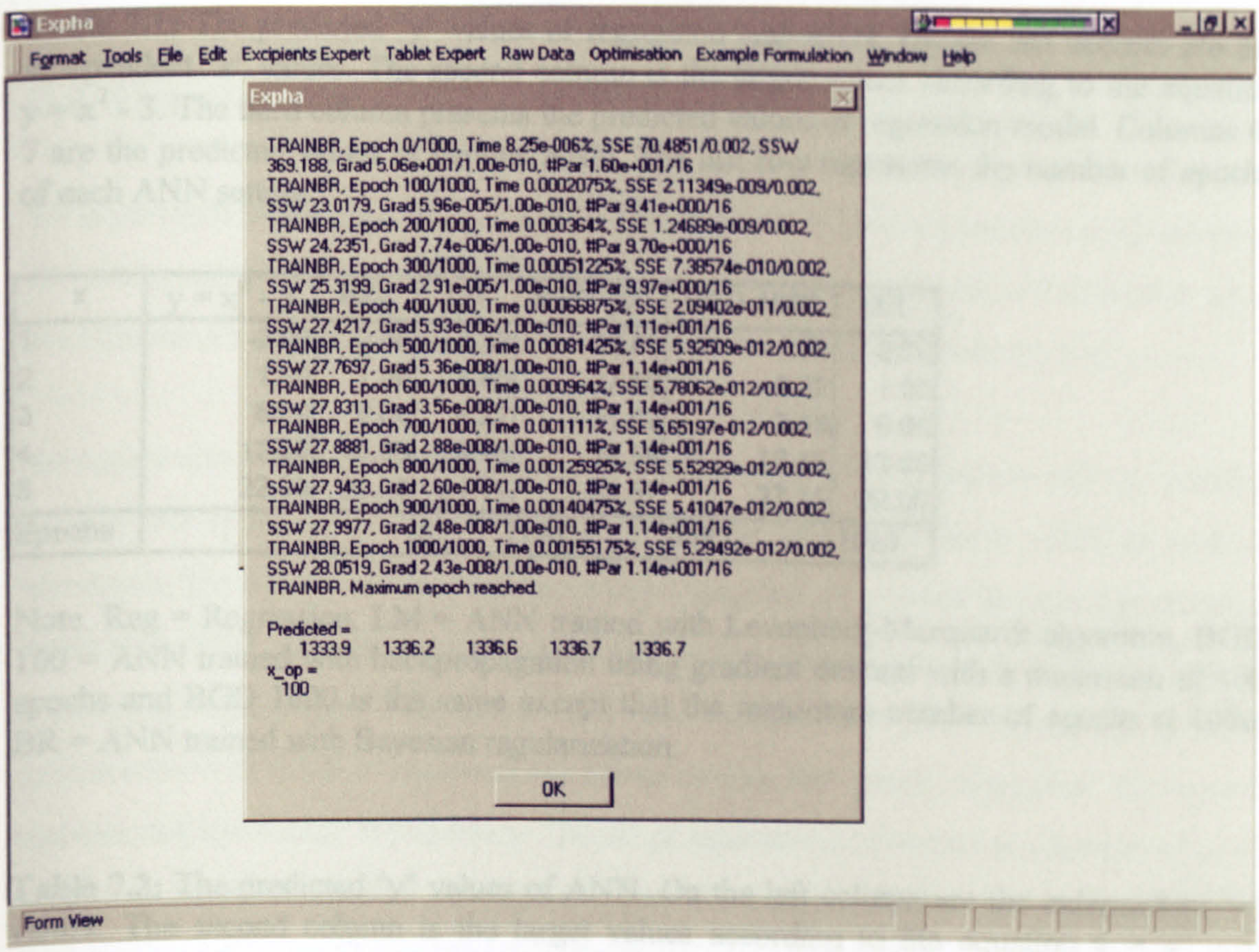


Figure 7.27: Final part of Expha output after pushing "Run ANN" button when ANN module in Expha was trained on data from the equation $y = x^2 - 3$. The training algorithm is ANN trained with Bayesian regularization with a maximum of 1000 epochs. "Predicted" are the predicted responses for independent variable *x* values of 60, 70, 80, 90, 100. "x_op" is the optimised value of the input variable which minimise the dependent variable.

Table 7.1: The predicted "y" values of regression and ANN. On the left column are the independent "x" values. The second column is the target values according to the equation $y = x^2 - 3$. The third column presents the predicted values of regression model. Columns 4-7 are the predicted values of various ANN. The last row represents the number of epochs of each ANN setting.

x	$y = x^2 - 3$	Reg	LM	BGD 100	BGD 1000	BR
1	-2.00	-2.00	0.32	22.65	-1.78	-2.00
2	1.00	1.00	1.85	27.50	-0.22	1.00
3	6.00	6.00	4.27	31.02	2.48	6.00
4	13.00	13.00	9.98	38.56	10.45	13.00
5	22.00	22.00	21.19	53.57	27.11	22.00
Epochs			3	100	859	1000

Note. Reg = Regression, LM = ANN trained with Levenberg-Marquardt algorithm, BGD 100 = ANN trained with backpropagation using gradient descent with a maximum of 100 epochs and BGD 1000 is the same except that the maximum number of epochs is 1000. BR = ANN trained with Bayesian regularization.

Table 7.2: The predicted "y" values of ANN. On the left column are the independent "x" values. The second column is the target values according to the equation $y = x^2 - 3$. Columns 3-11 are the predicted values of various ANN. The last row represents the number of epochs of each ANN setting.

x	$y = x^2 - 3$	GDM	GDX	OSS	BFG	CGF	RP	CGP	CGB	SCG
1	-2.00	-1.99	-3.65	2.52	-1.51	0.67	-6.16	-1.85	-0.13	-1.08
2	1.00	-0.44	-2.41	3.61	-0.36	1.89	0.41	-0.7	0.95	0.02
3	6.00	2.37	0.4	6	2.34	4.59	13.05	1.96	3.67	2.59
4	13.00	10.64	8.95	13.21	10.53	12.74	21.69	10.03	12.01	10.45
5	22.00	27.6	26.47	27.742	26.94	29.08	25.64	26.23	28.69	26.35
Epochs		803	87	8	6	8	56	8	5	6

Note. The abbreviations stand for ANN training method. GDM = gradient descent with momentum. GDX = gradient descent with adaptive learning rate. OSS = one step secant method. BFG = BFGS quasi-Newton method. CGF = Fletcher-Reeves conjugate gradient algorithm. RP = resilient backpropagation. CGP = Polak-Ribiere conjugate gradient algorithm. CGB = Powell-Beale conjugate gradient algorithm. SCG = scaled conjugate gradient algorithm.

Optimisation using the regression model yielded a suggested x value of 0 for the minimum point whereas optimisation of all 3 ANN models, that were best optimised (the 3 ANN training methods of columns 4-6 in Table 7.1), yielded a minimum x value of 9.

7.6 Discussion

An expert system has been created that has the capacity to help formulators at all levels of experience achieve their goals. The Expha expert system possesses basic formulation rules, has facilities for data acquisition with tools for data modelling and optimisation.

An expert system was created that is very practical for data collection and for building models and optimisation. It is unique in the sense that it combines ANN as well as regression. The latter gives an advantage since regression may model less complicated data better than ANN may. A unique combination is also created between ANN and type of optimisation technique—two areas that are traditionally not connected. Usually, optimisation with ANN is used with the newer models like genetic algorithm. The system incorporated knowledge in pharmacy. The set of rules could have been much better if more experts in tablet formulation development were involved in its development stage. With even more rules it would still have kept the same basic structure of an expert system. The latter limitation prevents the system from becoming the sole solution provider for tablet formulation problems. In a pharmaceutical company with a number of experts for tablet formulation it is not likely that Expha's advice will be needed. Formulation experts could use Expha for processing data, doing modelling and optimisation. The latter part is a strong aspect of Expha since it has extensive modelling techniques and the user can change many parameters in that area. Expha could be beneficial if used by pharmacist or other scientists that need educational guidance in the field of tablet formulation. Such people can be found in the schools of pharmacy and in the pharmaceutical companies. The section to be followed will discuss testing and field trial of Expha.

Figure 7.17 shows the regression equation Expha calculated. The equation that Expha tried to find the appropriate parameters is:

$y = a_{(0)} + a_{(1)} * x + a_{(2)} * x^2$.

After the training process Expha yielded to following values for the coefficients $a_{(1)}$, $a_{(2)}$ and $a_{(3)}$ respectively: -3, 0 , 1. Hence, the linear term of the equation became redundant to yield the equation:

$y = -3 + x^2$.

It is obvious that the regression model has learned the data successfully since it generated

the appropriate equation. In accord with that, Expha predicted the appropriate *y* values relevant to the *x* values of the prediction set. The optimised *x* value was calculated as 3.3×10^{-17} which is very close to the desired x minimum point value (theoretically, the minimum point should be zero).

The overall structure of the code was the same as was used in previous chapters. The underlying ANN structure was the same, no attempt was to optimise the topology since examples of doing so were already demonstrated in chapters 3, 4 and 5. This explains why one that selects different algorithm gets different answers. Obviously, the topology selected is inappropriate for the problem selected. A more appropriate topology with fewer hidden neurons should give similar predictions using the different learning methods but not identical because of the possibility of each method converging to a different local minima.

Figure 7.18 shows ANN trained with backpropagation using gradient descent with a maximum of 100 epochs. This ANN succeeded in learning the data and to lower the MSE as the number of epochs increased. The prediction results of Figure 7.19 show that the predicted values are quite far from the desired ones. Hence, it was decided to increase the number of epochs to 1000. Figure 7.20 shows it was right to continue training since the MSE kept lowering till it arrived at the target value of 0.002 after 859 epochs. Figure 7.21 presents much better predictions (see Tables 7.1/7.2 for the desired *y* values) than the ones arrived at with 100 epochs of training (Figure 7.19), this is in accordance with the differences in the MSE shown in graphs 7.18 and 7.20. Till now the learning patterns of the graphs and the improvement in the predictive ability as the number of epochs increases are standard and typical to simple backpropagation, so there is no need to suspect there is a flaw in the program regarding this learning method. Since 1000 epochs was more than enough for simple backpropagation to arrive at the desired MSE it was decided it would also be enough for other training methods that learn faster, with less iterations. Figure 7.22 shows a typical fast algorithm that arrives at the solution in only three epochs. In this case, this is the Levenberg-Marquardt algorithm and its predictions are presented in Figure 7.23. Table 7.1 summarizes training results for the models presented earlier. Table 7.2 shows other ANN models and typical learning behaviour of ANN can be seen; the more simple learning methods, like simple backpropagation or simple variations of this training method like momentum, need more epochs to arrive at the target MSE value relative to the more sophisticated ones. This behaviour is typical and was shown in studies presented in earlier

chapters and suggests the ANN training algorithms work correctly. Looking more closely at the speed of the algorithms, by inspection of the training results in Table 7.1 and 7.2, one can see that the fastest ANN was the one trained with the Levenberg-Marquardt algorithm. The conjugate gradient algorithms (Hagan et al., 1996) and the quasi-Newton (Dennis & Schnabel, 1983) learned the data in about the same number of iterations, they both needed less than 10 iterations. This order of learning algorithm speeds is in accordance with the literature (Demuth & Beale, 1998) as discussed earlier in section 7.4. The one step secant method (Battiti, 1992) also learned the data in less than 10 iterations. The quasi-Newton algorithm requires more storage and computation time than the conjugate gradient algorithm. The one step secant method is an attempt to bridge the gap between these two methods. This method is considered as a compromise between full quasi-Newton algorithm and the conjugate gradient algorithm. As such it is not surprising it has the same levels of speed as the latter two training methods. Looking in Tables 7.1 & 7.2 one can see that the fastest algorithms are not the best ones in terms of predictive ability. The learning algorithm with the best predictive ability was the Bayesian regualization ANN (MacKay, 1992). In this ANN the weights and biases are assumed to be random variables with specified distributions. The regularization parameters are related to unknown variances associated with these distributions. These parameters are estimated using statistical techniques. This algorithm for training ANN provides a measure of how many ANN parameters (weights and biases) are being effectively used. Looking in Figure 7.24 on the bottom plot (number of parameters is on the y-axis and number of epochs is on the x-axis) one can see the ANN uses approximately 11 parameters. This type of ANN eliminates the guesswork required in determining the number of neurons in the hidden layer, since no matter how large is the number of parameters in the ANN, the effective number of parameters should be the same. Hence, Bayesian regualization ANN avoids the problem of overfitting by too many parameters in the ANN.

In the optimisation process there is a feedback mechanism between the model, whether it is ANN or regression one, and the optimisation algorithm. In this case an arbitrary value of 100 was the starting point from which the optimisation algorithm queried the model as to the predicted response (y) and according to input from the model the optimisation algorithm decided on the next guess and so on. Figure 7.25 shows that the Bayesian regularisation model succeeded in predicting the five y values accurately but failed in the optimisation process since the final guess was the optimisation starting guess. It was suspected that

there might be flaw in the optimisation routine. Another possibility was that the ANN model did not succeed in generalising the data and just memorised the data. To check the generalisation ability the model was queried with values it was not trained on. From Figure 7.26, the model predicted accurately the responses for the independent variable x values between 0.5 to 19.5 in increments of 1 (0.5, 1.5, ...19.5). Once the possibility for lack of generalisation ability was eliminated it was decided to check the extrapolation ability. The model was queried about the y values of the following independent variable x values: 60, 70, 80, 90, 100 and the responses it generated are given in Figure 7.27, they are 1333.9, 1336.2, 1336.6, 1336.7, 1336.7 respectively. Hence, the lack of optimisation ability is because the model could not extrapolate and not because of a bug in the optimisation algorithm.

In order to help in the selection of the appropriate training algorithms, basic descriptions and highlights of Expha ANN training algorithms are given in this section and will also be incorporated with Expha.

Basic gradient descent - the simplest backpropagation training algorithm. It updates the weights in the direction of the negative of the gradient. All the calculations of the derivatives (e.g. for calculation of the gradient) in all the training methods described in this section are with respect to the weights and biases. Detailed example of a calculation of feedforward and backward pass with this training method was given in the Background. It is slow training method.

Gradient descent with momentum - this is the same method as previous one with added momentum term. The momentum allows the ANN to respond not only to local trends in the error surface but also to recent trends in the error surface. The momentum can help in avoiding local minima. This method is often faster than basic gradient descent and it too is explained in the Background.

Adaptive learning rate - the performance of the algorithm is highly sensitive to the adequate setting of the learning rate. If the learning rate is too high the algorithm became unstable. If it is too low it will take long time for the algorithm to converge. The solution is in the form of variable learning rate that adapts its size according to the error surface. This training method is faster than basic gradient descent and it is also explained in the Background.

Resilient Backpropagation (Riedmiller & Braun, 1993) - ANN usually use sigmoid transfer functions as activation functions. In this thesis two types of these functions were used (in the neurons of the hidden layer), one outputs value in the range of 0-1 and the other -1-1.

These functions are 'squashing' functions since they compress an infinite input range to a finite output range. Hence, the gradient could have a small value so it will cause only minor changes in the weights and biases although much bigger changes are needed. The aim of this training method is to eliminate this destructive learning effect caused by the size of the partial derivatives. It does so by taking into account only the sign of the derivative to determine the direction of the weight update. The magnitude of the weight change is determined by a separate parameter. If the derivative is zero there is no change in this parameter. Whenever the weights are oscillating the weight change will be reduced. If the weight is changed for several iterations in the same direction than the weight change will be increased. This algorithm is considered a fast one.

The following algorithms are based on conjugate gradient algorithms (Hagan & Demuth, 1996). The backpropagation algorithms adjust the weights in the steepest descent direction that is the negative of the gradient. This can cause the most rapid decrease in the error function but it does not necessarily produce the fastest convergence to a predefined MSE. In conjugate gradient algorithms a search is performed along conjugate directions. The learning rate determines the size of the weight update also termed as the step size. Usually, in a conjugate gradient algorithm the step size is adjusted in each iteration. The search for the optimal step size is done in the conjugate gradient direction (step size that will minimize the error function). In the first iteration there is a calculation of the negative of the gradient.

$p_0 = -g_0$

Where g_0 is the gradient in the first iteration.

Then there is a line search to determine the optimal distance to move along the current search direction.

$x_{k+1} = x_k + \alpha_k p_K$

Where x_k is a vector of current weights and biases and α_k is the learning rate.

Then the next search direction is conjugate to previous search directions. The new steepest descent direction is combined with the previous search direction.

$p_k = -g_k + \beta_k p_{k-1}$

Where β_k is a constant. The manner in which this constant is calculated differentiates the different conjugate gradient algorithms.

The conjugate gradient algorithms are usually much faster than gradient descent with variable learning rate and sometimes faster than resilient backpropagation.

In the Fletcher Reeves conjugate gradient algorithm (Hagan & Demuth, 1996) the β_k constant value is the ratio of the norm squared of the current gradient to the norm squared of the previous gradient.

In the Polak-Ribiere conjugate gradient algorithm (Hagan & Demuth, 1996) the β_k constant value is the inner product of the previous change in the gradient with the current gradient divided by the norm squared of the previous gradient. It is impossible to say which one of these two conjugate gradient algorithms is better and their performance is considered similar.

Powell-Beale conjugate gradient algorithm (Powell, 1977) differs from other conjugate algorithms in the reset point of the search direction. The conjugate gradient algorithms reset the search direction to the negative of the gradient (see in the first iteration described earlier) when the number of iterations is equal to the number of weights and biases. The criteria of resetting the search direction according to Powell-Beale conjugate gradient algorithm is

$$|g^T_{k-1} g_k| \geq 0.2 \|g_k\|^2$$

Where g_k is the current gradient. This expression checks if there is orthogonality left between the current gradient and the previous one. There are problems in which this algorithm is preferable on the Polak-Ribiere conjugate gradient algorithm. But it is difficult to predict a-priori on which problems it is better.

Scaled conjugate gradient algorithm differs in that it is not performing a line search procedure that requires extensive computations. Each one of the conjugate gradient algorithms performs line search. A line search procedure evaluates the error goal for a given number of points, e.g. starting at certain distance and doubling it in each step along with the search direction. When the error goal increases between two successive points than the minimum lies in the area between these two points. The size of the interval is then reduced and two new points are located between this minimum interval. The values of these points determine a section of the interval that can be discarded and a new point is placed within the new interval. This procedure is repeated till certain interval value is reached. This line search algorithm is called Golden Section Search (Hagan & Demuth, 1996) but there are many other line search procedures. The line search requires the ANN to compute the responses to all training inputs several times for each search. Moller (Moller, 1993) developed this Scaled conjugate gradient algorithm, that doesn't use the line search, in order to save computation time. This training algorithm may requires more iterations than the other conjugate gradient algorithms but the number of computations in each

iteration is much lower (less time to compute each iteration).

The following two algorithms that will be described are Quasi-Newton algorithms. Newton's method is an alternative to the group of conjugate gradient algorithms. It is based on the following equation

$x_{k+1} = x_k - A_k^{-1} g_k$

Where x_k is a vector of current weights and biases, g_k is the current gradient and A_k is a matrix of the second derivatives of the error function. This matrix is called Hessian matrix. In comparison to the conjugate gradient group this method often show better results but it is much more time consuming because of the need to compute the Hessian matrix. As an attempt to bypass this problem of computation a group of algorithms called Quasi-Newton Algorithms was developed that approximate the Hessian matrix by the use of the gradient.

BFGS algorithm (Dennis & Schnabel, 1983) - this quasi-Newton algorithm is considered the most successful one. BFGS are abbreviations for Broyden, Fletcher, Goldfarb and Shanno. This algorithm will usually converge faster than the conjugate gradient algorithm but it requires much more computation time at each iteration. As the number of weights and biases became larger it became much more time consuming to calculate the approximation of the Hessian matrix. Hence, this method is recommended for small ANN. For the larger ANN (with more weights and biases) it is recommended to use resilient backpropagation or one of the conjugate gradient algorithms.

One step secant algorithm (Battiti, 1992) - this algorithm is considered a compromise between the full quasi-Newton (also called secant) algorithm and the conjugate gradient algorithm. It assumes that at each iteration the previous Hessian was the identity matrix. In addition to memory saving by doing so it also eliminates the need to compute a matrix inverse for the new search direction. The computation time is less than needed for the BFGS algorithm but more than the conjugate gradient algorithms.

Levenberg-Marquardt algorithm (Hagan & Demuth, 1996) - this method also tries to approximate the Hessian matrix. If the error function that is to be minimised is the SSE error function as all the ANN used in this thesis then the Hessian matrix can be approximated as

$H = J^T J$

Where 'J' is the Jacobian matrix which contains the first derivatives of the ANN errors. It is much simpler to compute the Jacobian matrix then to compute the Hessian matrix. The

gradient can be calculated as

$g = J^T e$

where 'e' is a vector of the ANN errors. The basis for the Levenberg-Marquardt algorithm is the following equation:

$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e$

Where x_k is a vector of current weights and biases. Newton's method described earlier is best used near the error minimum. In big μ values the training method became gradient descent with small step size. When μ equals zero the training method became Newton's method with approximation of the Hessian matrix. Since the aim is to move towards Newton's method - if the SSE decrease there will be reduction in the μ value and if the SSE increase there will be increase in the μ value. As was mentioned before, in ANN of up to several hundreds weights and biases (moderate size ANN) this training method is considered the fastest one.

Bayesian regularization - this training method tries to minimise both the SSE and the number of weights (and biases) to generate an ANN topology (small as possible) that has good generalisation ability. The update of the weights and biases was done in Expha with the Levenberg-Marquardt algorithm (Bayesian regularization with Levenberg-Marquardt algorithm is discussed in Foresee & Hagan, 1997). The advantage of this ANN was demonstrated earlier (Table 7.1 demonstrates that it was the only ANN training method that could predict accurately the simple function) in that it eliminates the problem of inappropriate topology selection by the user.

As was mentioned in section 7.4.3 (below Figure 7.12) there is not one definite ANN training method that is the best and should be used in all cases. It was also mentioned that the speed comparison of the algorithms is important, a subject that was also addressed previously in this section since the speed of the algorithm to converge to a desired MSE is very important property of the algorithm and one of the factors that should be considered when selecting one. Speed is even more critical in view of the fact that in small number of experiments (that is usually the case) the most robust way to validate the model is the leave-one-out, which is very time consuming. Hence, it is recommended (in my view) to begin with the fastest algorithm that is the Levenberg-Marquardt algorithm. If the validation results are not satisfactory then go to the Bayesian regularization method which saves the time of finding adequate methodology. If either of the methods is not giving satisfactory validation results then try the other training algorithms. In any case it might be

useful to the user to read the short descriptions of each training method presented earlier (it will later be incorporated into the help menu of Expha), since they not only relate to training speed but also to other parameters like size of ANN and complexity of calculations. To avoid overfitting it is important to note that it might be useful to follow Kolmogorov's theorem (in all ANN training algorithms described earlier except Bayesian regularization) which states that twice the number of independent variables plus one is enough hidden neurons to model any function. Tables 7.1 and 7.2 show the usefulness of this guideline since it demonstrated inability of ANN that did not follow this guideline (the ANN used should have maximum of $2 \times 1 + 1 = 3$ hidden neurons so 5 hidden neurons that were used lead to too complex ANN) to properly learn a simple equation.

Since the expert system does not currently give details as to how one goes about developing an optimised formulation, the general approach to developing an optimum formulation is given below and will be at some stage be incorporated into the help menu of Expha. First, it is suggested looking in the "Example formulation" menu that has formulation examples from "Pharmaceutical Dosage Forms Tablets" (Lieberman, 1989) and "Physicians Desk Reference" (published by Medical Economics Company, Inc., 1997) which also has data on formulations but without the quantities of the excipients used or the process involved. One can also look in the "Select Excipients" form to see formulations that were already made in the past by the organization where the Expha was being used. There are other facilities that can give the user basic idea regarding the formulation like the "Advice" button or from the recommended excipient concentrations. The user then decides on which parameter he will focus and then implement an experimental design e.g. full factorial as Patel (Patel, 1996). Alternatively, he can enter into Expha formulations that were done in the past (which may or may not have followed an experimental design) with the aim to get optimised formulation on the basis of this limited data.

Expha enables all the experimental variables to be stored for all the experiments as well as the measured properties. This data can then be used to develop a model in Expha using either regression or ANN and then to predict the optimum formulation. There are two approaches to optimization: static and dynamic or sequential. With the static approach one defines number of experiments at the beginning. Typically this would employ a full factorial design as in Patel's data (Patel, 1996) and then one fits in regression a polynomial equation. In ANN one would train the net as was done in earlier chapters in this thesis.

Then one should validate the model and optimise to get the optimum formulation. The second approach is sequential and is based on the simplex method. Here you carry out a starting set of experiments and then the simplex identifies the next experiment and you carry on in this way sequentially. This approach is not yet implemented in Expha.

The process of building models and optimisation is the most sensitive (prone to programming errors) since it involves transferring data to the MATLAB® software, which acts as a calculation server to Expha, and receiving the calculated values from MATLAB®. Hence, correct communication between the programs is essential for success in the calculations. The algorithms for modelling and optimisation work correctly, in this case communication being essential between the two programs-the MATLAB® programs of ANN/regression models and the optimisation routine. The success of the modelling and optimisation algorithm was verified by looking at the patterns of learning and optimisation and checking that they were reasonable.

In this section comparison between Expha and other expert systems will be made. As opposed to Lai et al. (1996) capsule formulation expert system that contains many rules, Expha contains only few rules. As in Expha expert system, Lai and her co-workers (Lai, et al., 1996) utilised a database of excipients. Formulations are deduced automatically, whereas Expha only aids in formulating. In Expha one can see all the knowledge and in the capsule expert system there is no access to the database since sometimes, proprietary knowledge has been used. The programming language that Lai et al. (1996) used is C. The programming language used to program Expha is Visual Basic® for Applications and MATLAB®. Visual Basic® for Applications® was used to program everything connected to the database programming such as rules applied to the data and also for programming the user interface. MATLAB® was used for the computations involved in building models and optimisation routines. The user interface in Expha is more user friendly because a more suitable programming language was used for that purpose. Lai et al. (1996) stated that the expert system is never completed since the user enters new knowledge all the time. The same is true for Expha since the user in Expha can enter more excipients into the database or enter new information about current excipients.

Expha was not built on an expert system shell (as was done by Rowe & Upjohn, 1993a) but was built with programming languages allowing the developer more flexibility. Expha

also used ANN but used optimisation techniques different from genetic algorithm. ANN and genetic algorithms are incorporated into Cad/Chem, which is software used to build models and optimise formulations (Colbourn & Rowe, 1996). Expha also contains powerful model building and optimisation facilities. However, it also contains inherent database and expert knowledge as opposed to Cad/Chem. This unique combination is not used in any expert system.

In the future Expha will also integrate a tablet coating phase, as opposed to the tablet coating expert system (Rowe & Upjohn, 1993a) that is a stand-alone expert system for solving tablet coating problems. The reason these two domains of tablet formulation and tablet coating were integrated into Expha stems from the concept that tablet coating is influenced by factors related to the formulation (i.e. to the core tablet). As an example of the latter issue, formulating with a lubricant with a melting point of 50°C and using 60°C in the tablet coating process will mean pits in the tablet surface due to lubricant evaporation from the tablet surface.

As in Expha, in Zeneca expert system (Rowe & Upjohn 1993b) for tablet formulation drug properties are entered into the database. It selects the excipients that are characterised by their functional category groups, as in Expha. The difference is that Expha does not select the excipients automatically. Expha gives advice regarding the formulation excipients such as the inclusion or exclusion of certain functional category groups, for example sometimes it will state there is no need to select disintegrant. The advice is also in the form of information regarding excipients, e.g. the recommended concentration of the excipient, a factor that is influenced by the role of the excipient in the formulation (the same excipient could have several recommended concentrations depending on its role in the formulation). All the functional category groups that are included in Zeneca expert-system are present in Expha but Expha also contains an antihadherent group which has the role of preventing sticking to the punch and the die wall (Lieberman et al., 1989). Both expert systems include a database on excipients. In Zeneca expert system as in Expha the user enters the measured tablet property results like disintegration time etc. In Zeneca expert system (Rowe & Upjohn 1993b) there is an iterative process of suggesting the formulation, comparing it with the result and suggesting a new formulation (an on-line process). The parallel stage in Expha is collecting the input variables that could dictate, for example, several excipient concentrations in the formulation. Followed by collecting the measured properties of these

tablets with the different formulations. Then there is a model building process and according to the user constraints and goals, the computer suggests the optimised tablet formulation parameters.

Regarding other expert system; in the Cadila expert system for tablet formulation (Ramani et al., 1992) the user enters drug properties as in Expha. As opposed to Expha there is no process of formulation optimisation.

The difference between Expha and other systems is that Expha is an open system that will be published. Expha is not a proprietary product. It will be useful for other academics to use it as a basis and to extend this work in AI and expert systems further. The next section will discuss the future prospects of Expha.

MATLAB® is an object-oriented language (Breiner, 1999) that was chosen carefully to be the environment for the intensive computations for ANN, optimisation and also for simple, fast regression calculations. The reason for this choice is explained with an emphasis on the future benefits.

MATLAB® is much easier to program than other object-oriented languages like C++. C++ forces one to delve into programming and deal with messy issues like pointers, a subject that relates to the storage of data in computer memory. In MATLAB® the programmer just thinks about the algorithm. After knowing logically how to solve a problem it is very easy to implement it in MATLAB® language. So any further developer of Expha will easily understand the code and develop new algorithms without any problem. There is also the benefit of a language that is easy to maintain.

MATLAB® programs have been run on a cluster of PC's (Boskovitz et al., 1999). That means, it is possible to run an ANN simulation on several PC's to reduce training time. This time saving is especially important if one wants to use the leave-one-out method as the validation method. MATLAB® has capabilities of data acquisition. It can communicate with a variety of off-the shelf PC-compatible data acquisition hardware (St. John-Olcayto, 1999). It means MATLAB® can read in data from machines like a tablet coating machine and pass parameters like inlet air temperature directly to the database in Expha.

Implementation of the tablet coating process has begun. The concept behind this is that tablet coating is influenced directly by the formulation and by tablet properties. The level of lubricant like magnesium stearate has an influence on the tablet contact angle, so it will influence the strength of adhesion of the coating layer to the tablet surface. Lubricant properties like melting point should be taken into account, the following example (mentioned earlier) demonstrates this. Stearic acid has a melting point of 54°C, when coating with inlet temperature of 60°C one can see holes in the tablet coating since the lubricant evaporates from the tablet coating layer. Another tablet coating problem is that with a hydrophilic drug tablet one has to choose a hydrophilic coating to allow hydrogen bonds with the tablet surface to be created. Otherwise, tablet surface and coating layer will repel each other. If a tablet property like friability is high it will cause coating problems. This is because in the coating process debris of powders will be created which sticks to the tablets, thus causing uneven 'orange peel' like coating. Another reason is that the coating material has difficulties to stick to the tablet because of abbrasion. The key to solving these types of problems is selecting the appropriate fields and to invoke according to values assigned to these fields appropriate routines, like a message box that advises the user regarding a specific situation. So the first step is to have the appropriate fields in the database. "Tablet Properties" form has to have friability (from the tablet coating point of view it is the most important tablet property since it influence the adhesion of coating layer to the tablet surface), excipients have to include fields for their melting point, and if they are hydrophilic or hydrophobic. Tablet coating process parameters, like inlet air temperature, have to be included. After putting in the appropriate fields, an algorithm for invoking the appropriate action needs to be created. It could be a simple one showing only a photograph of the result of lubricant going out of the tablet coating (i.e. show impaired tablet coating) if the lubricant melting temperature is lower than inlet air temperature. Hence, it is obvious there is an advantage utilising the current database to implement a tablet coating stage.

7.7 Conclusions

The expert system developed is a basic tool that can be a starting point for people new in the area of tablet formulation. A number of pharmaceutical rules where implemented, they help in process selection or in giving advice related to other areas. The data itself is also useful for formulation, e.g. when an excipient is being chosen, Expha will present minimal and maximal concentrations that can be used as a guideline for the user. The parts of model building and optimisation could be practical also for experienced formulators. In that respect, there was emphasis on the implementation of many training methods for the ANN and not just using simple backpropagation (it gives an advantage since different training algorithms could have different learning ability for the same data, as demonstrated in Chapter 4). In addition, there are various other parameters of the ANN that can be changed like ANN topology or criteria to stop training. All parts of the expert system were validated. In that respect special emphasis was given for the modelling and optimisation parts. In addition, the parts of data input and output like adding more excipients and retrieval of excipients' data or entering/retrieval of formulations were extensively tested and proven to work correctly.

The system incorporates new elements that are not available in other expert systems. The system is built in a way that enables extension easily in two main paths. One path is the possibility to add more modules like module for tablet coating with its relevant rules. The second area that could be expanded is the addition of more rules, in that respect the problem is not a programming one but the data mining from the experts that could be used in the form of rules and implemented in the expert system.

To summarise, Expha, a suitable tool for novice formulators was created. Expha can also be very useful for experienced formulators. The variety of features, the possibility to use it for data collection throughout the formulation development process, gives it a good chance of being useful for many formulators. Hopefully, it will broaden the knowledge of formulation development for many formulators and develop their creativity and ability to find adequate solutions for difficult formulation problems.

8. General Discussion & Closing Remarks

This concluding chapter looks again at the debate about superiority of ANN over traditional modelling methods. Some conclusions arrived at as a result of this work are presented. There is a discussion of the issue of the ratio of the number of data points to number of unknown parameters (composed of bias and weights) in ANN. It also shows how the regression modelling in this thesis is different from previous studies and how the comparison between ANN and regression differs from other studies. Issues such as topology selection and various other aspects from Chapters three, four and five will be discussed, tackling problems like whether it is preferable choosing ANN with output neurons representing all responses or just one output neuron representing only one response at a time. The optimisation routine employed on the tablet data (Chapter 6) will be discussed briefly. Expha expert system (Chapter 7) will be presented very briefly with respect to several summarising issues. These include answers to questions like - how Expha is different from other expert systems and what it is good for. Then difficulties in the creation of Expha follows and its future aspects are discussed.

The debate—which method is better to model data, regression or ANN, is a subject that researchers in many fields are interested in. Veng-Pedersen & Modi (1992) wrote an article with the provocative title: "Neural Networks in Pharmacodynamic Modelling. Is current Modelling Practice of Complex Kinetic Systems at a Dead End?" One of the concluding phrases of this article was *ANN offer a challenging empirical based alternative to any*

complex multivariate kinetic system modelling aimed at predictions... This article brought criticism by Siegel (1992) who stated that ANN do not give insight to the science, they can only be used as a tool to predict various phenomena. If scientists had this tool a long time ago it might be that since they could predict with ANN they would not bother to develop theories that would lead to building equations (used for predictions) describing the system under investigation and make progress in science. Hence, ANN can impair scientific progress if used instead of old methods. Siegel also had technical comments regarding the study presented in the article. Siegel noticed that ANN extrapolated in an incorrect manner, that there is a bias in ANN predictions since most of the ANN predicted values were above the observed values. Veng-Pedersen (1992) replied to Siegel's criticism and stated that ANN will not replace the current modelling techniques but when traditional modelling methods do not succeed ANN would be in use. It seems Veng-Pedersen was withdrawing his statements.

In accordance with Siegel's opinion (1992) that statistical methods of inspecting model predictions should be used, this thesis used statistical techniques in Chapters four and five to analyse ANN and regression performance. In spite of the fact that we do not fully understand ANN it is not true that they are complete black boxes and that they do not give insight to the system. There are techniques that help us understand what happens inside ANN—like Hinton diagrams (Demuth & Beale, 1994) which illustrates the size of the weights between neurons (the strength of neuron's connections). Other techniques that give insight into ANN will probably be developed in future. ANN may predict phenomena we do not fully understand but as ANN are better understood the phenomena that ANN can model would be better understood.

With many of the studies reviewed in the Introduction, sometimes the models chosen after validation phase are models with more weights than data-points. Probably the studies also used bias terms (not always mentioned if bias terms were used or not) that can be considered as weights with a constant input connected to them. There is a contradiction between these results and the statement made in the Introduction (Hagan et al., 1996) that when there are more weights than data-points it should cause overfitting. Another point that could be seen in different studies and also in this thesis is that overtraining can cause overfitting. It can be avoided by carefully monitoring the number of iterations as was done in this research in the tablets and capsules study (clearly demonstrated in the capsule

study). These questions with ANN suggest similar questions for regression such as, what is the parallel in regression to overtraining and is it possible in regression to build a model for which the number of unknown parameters is greater than the number of data points? These questions not being valid suggests that regression and ANN are two different methods in essence even to the mathematics behind them. In a personal communication with Prof. Chris Bishop at the ANN summer school at Cambridge in 1997, he expressed the opinion that one could not say as a rule that overfitting would occur if ANN had more weights than datapoints. Only a validation step will assure the model is adequate. There are examples of ANN with much more weights than the number of data points and yet these ANN have good generalisation ability. Fewer weights than number of data points seems to be more a guideline than a rule.

This thesis reviewed studies like the one of Hussain et al. (1991) that not only did not optimise the regression model using methods like stepwise regression but also did not try several models (like second order etc.). Add to this the selection of the longest equation possible with the same number of parameters as the number of data points. They compare this regression model with ANN and stated that for their data ANN is the superior modelling method. As opposed to the latter type of studies it was shown in this study that regression prediction performance could be improved by trying different models and employing methods to optimise the regression equation. The comparison between methods aimed to be as comprehensive as possible and to survey several aspects and not just to examine if ANN has lower MRE than regression like Murtoniemi et al. (1994a & 1994b). Additionally, as opposed to other studies, the MRE comparison was done by taking into account all the predictions of the validation sets (in the tablet's data it was composed of 27 predictions for each response) of ANN and regression to test whether ANN was statistically better than regression. Looking only at the MRE can be misleading. From the data of Ebube et al., (1997) the MRE of the eleven ANN predictions (of percent drug dissolved after one hour) calculated, using the leave-one-out for validation, resulted in a big MRE may lead to the conclusion that there is no predictive ability. The reason for this is the very big percent relative error in trying to predict the first data point relating to the slowest dissolved tablet: (11.7-0)/11.7×100=1170% relative error. In this data point, polymer 2 fraction was in its lowest value and polymer 1 fraction was in its highest value (they are the independent variables). This reveals a common problem of ANN, that is, the inability to extrapolate. Hence, for both studies of tablets and capsules, all percent relative error results

from the best ANN and regression models were presented (to see if a few data points had a major influence on the MRE) and appropriate tests were conducted on these sets.

In the introduction of this thesis (section 1.7.1.1) it was shown by examining Murtoniemi et al. (1994a) MRE data that even when MRE seems to be small it still does not show whether a successful predictive model was built. As a predictor to the Murtoniemi et al. validation set, the average values of the granule size response were taken. Computing the MRE values of the validation set with this predictor gave better results than Murtoniemi et al. ANN/regression models. The suggestion of a predictive capability came from the fact that the data did not have much variability so the MRE would be low even if only the average value is taken as predictor. This study admitted it did not succeed in modelling all the responses in the capsule study.

Hussain et al. (1994) trained ANN on all the data as a method of selecting the best topology. In both studies of the tablets and capsules it was not possible to predict the optimised ANN (giving the best validation results) topology by training ANN on all the data. It was concluded that it is not possible to save computer time by omitting the phase of validation experiments on all topologies to choose the best one.

The selection of ANN with one output neuron to present only one response (at a time) of the problem, instead of several outputs (to present all responses simultaneously) should be considered depending on the relation between the responses. In the capsule study disintegration time was modelled alone and gave better predictions than ANN trained with all responses. The tablets study showed the opposite. ANN trained with only the dissolution rate as a response showed inferior prediction ability to ANN trained with all eight responses. This suggests that although training with only one output gives an advantage of reducing system complexity it is also a disadvantage of loss of information. That loss of information stems from the relationship between responses. Dissolution rate response is more related to the other responses in the tablet study than the relation between disintegration time response and the other responses in the capsule study.

In the tablets study ANN predictions showed better correlation (than regression) to the observed values in the response of mean weight. Examination of MRE results show ANN predicts better the extreme values of mean weight and dissolution rate responses. ANN

showed lower MRE for the predictions of hardness, erosion friability and dissolution rate responses. However, this difference is not statistically significant.

In the capsules study the responses that show lower MRE of ANN predictions relative to regression predictions (using simple linear models) are the minimum and maximum bulk density and disintegration time. For disintegration time response, regression predicted values show higher correlation to the observed values than ANN predicted values. As opposed to the tablet study, which shows high correlation between ANN and regression predictions in all responses, in the capsule study there is no correlation between ANN and regression predictions in disintegration time response. The ANN offers a completely different predictive alternative in the modelling of disintegration time response. However, one complicated regression model that was built showed (by its lower MRE relative to ANN) that the ANN had lower MRE in predicting disintegration time response due to lack of flexibility of the regression model. This regression model that was built arbitrarily just to verify one of the conclusions of the study (after the study was completed) also took into account square and interaction terms that gave it more flexibility relative to the simple linear regression equation.

The improvement in ANN predictions, from using only simple backpropagation in Chapter 3 to Chapter 4 where different learning algorithms were employed, showed it is worthwhile not just examining different topologies and number of epochs but also trying different learning algorithms for ANN. The ANN did not show better predictions for limited data as opposed to Bourquin et al. (1998a), who showed ANN is much less sensitive than regression to the organisational level of a trial design. Nevertheless, the tablets and capsules study show that it is worth using ANN because in different respects of these studies it can give better predictions.

Had this study used just one regression model, as some studies did, the conclusions of this research would be that ANN is much better than regression for the prediction of solid dosage form properties from well designed experiments as well as not so organised experiments. However, this study gave a fair chance to regression and using various statistical techniques showed regression also has its benefits and its worth using this methodology.

In regression, after programming the computer tries several models of regression equations and checking their validity by the leave-one-out method, each time the program runs a model and validates it, this process takes less than one minute. In ANN this process of testing numerous models and testing them by the leave-one-out method is a time-consuming process. In fact, running different ANN models on the tablets and capsules data took several months for each one of the studies. In real life often there are time constraints, the formulator does not have too much time to build models (i.e. to train them). In this respect regression models are superior to ANN ones. Only RBF ANN, that proved to be the best models in part of the tablet's properties responses, are not much inferior than regression in the area of computation time. Testing RBF ANN with the leave-one-out method took only several minutes. Another similarity between regression and RBF ANN is that both of them, as opposed to ANN trained with backpropagation, yielded the same prediction results when the same specific model was trained for the second time. The difference is because ANN trained with backpropagation have different starting random weights each time it is trained. In using RBF ANN it is important to monitor carefully the spread constant since a change in it can influence the ANN prediction ability. RBF ANN of the type that neurons are added to the ANN until the sum squared error falls beneath an error goal or maximum number of neurons has been reached, showed better performance than the type where there are as many RBF neurons as there are input vectors. Hence, the number of neurons in RBF ANN is an important factor to be considered and the dynamic approach for selecting the number of neurons seems better than the rigid one.

The concept of combining optimisation method, which was developed before ANN methodology began to spread, with the new modelling technique was employed in Chapter 6 that optimised ANN trained on part of the tablet data. The ANN was trained only in relation to the responses of impact friability and disintegration time. The same optimisation routine could have been applied with all responses but only two were chosen for the ease of demonstration and explanation. The multiobjective optimisation algorithm of Gembicki (1974) was employed. It was shown that trying to start the optimisation with the data point that shows the best responses is not a necessary criteria and starting with an arbitrary point resulted in the optimum point with even less time, employing fewer optimisation iterations. It was also shown it is worthwhile using optimisation techniques since the optimised solution potentially had much better response values than the point of best responses selected from experimental results. However, only experimental results from a formulation

according to the optimised independent parameters could confirm the predicted responses of this optimum formulation. As the validation results of the ANN, used in the optimisation routine, are better there is more chance the predicted optimised formulation, attained by the optimisation routine, is reliable.

An expert system, Expha, was created with a wide variety of options for model building and optimisation using regression and ANN. The concept of trying various ANN training algorithms and various regression models was implemented in Expha. This unique diversity in backpropagation learning algorithms of ANN is not presented in any other software package according to article by Bourquin et al. (1997a), which reviewed current ANN software. Other learning algorithms like radial basis function ANN will be incorporated in the future, since it was shown they could predict better some of the responses and do this faster.

In Expha the user steps through the production process of collecting data that is used for building up models and optimisation. The system gives the user advice regarding the process and selection of formulation excipients. Algorithms for process selection were implemented using pharmaceutical knowledge necessary for its development. The selection of excipients employed both an excipients database and implemented rules. There are also rules of thumb and equations implemented in Expha. Expha's user interface creates an environment for creative formulation ideas and pushes the user to be more active in the field of tablet formulation.

Expha does not suggest a formulation like other expert systems, but guides the formulator (good also for a novice formulator) through the process of tablet formulation development, moving one in the direction of trying several formulations, building models with them and optimisation. Behind Expha is the concept of educating formulators not to be confined to database of formulation solutions they have in their company. Instead, trying new excipients, new formulation ideas with the old formulations for examples (accessed through the "Example Formulation" in the menu). Hopefully, this software will educate formulators not to give up if they fail to solve problems with methods that have been used to solve previous formulation problems. This software aims to encourage a creative process of solving tablet formulation problems.

To summarise, relevant articles in the field of solid dosage formulation were reviewed. New approaches to modelling both ANN and regression were presented. Robust comparison between the methods was conducted using various statistical techniques. Expha expert system was designed based on a new concept. Its limitations and future will be discussed in the subsequent sections.

Difficulties

The most difficult problems with the ANN/regression models were how to choose network parameters like topology, and how to check the generalisation ability. In relation to the latter problem there was also the problem of lack of data resulting in leave-one-out experiments that were time-consuming especially for backpropagation. Another problem was to resist being biased by previous articles which claim that ANN are better than regression to see objectively how regression and ANN stand relative to each other.

It was difficult to envisage how this expert system would look in the end. It was not designed by defining in advance the exact specifications of the product. The creation of Expha began by gathering many ideas that were thought to be relevant for tablet expert system. The next step was deciding how to implement these ideas into the system. Several times upon finishing the implementation of an idea a better way was conceived. This meant implementing the idea in a completely different way beginning from scratch and trying to fit the new ideas as good as possible to the target user, the pharmaceutical formulator. In summary developing the flow of the program and the concepts was the most difficult part.

For comparison with other expert systems there was attempt to gather information as much as possible on them. The best way of learning about software is trying it and looking at the documentation. But since these systems are very expensive it was not possible to finance the purchase of one. Even with the possibility to purchase one it is doubtful whether mention is made in the program itself or in the documentation how these systems are built which is of major interest to any researcher wanting to develop new concepts in formulation decision making. This problem was tackled by resorting to the literature (e.g. Rowe & Roberts, 1998) which was also lacking a lot of details that were of interest to this research. Nevertheless, the main concepts were included in the literature, which could be used as a source for comparison.

Finally, although the programming for the creation of Expha was done in computer languages that are considered easy ones, it was sometimes a difficult task. One of the main problems to resolve as a programmer was how to make the databases communicate with the optimisation routines.

Future

Regression is a reliable methodology, relatively simple to use with a history of many years of successful applications in pharmaceutical formulations field. Behind this method there is a robust statistical base. There has however been no significant evolution in this method and what was done 30 years ago by Schwartz et al., more or less is used by those using this methodology today. Formulators with an old generation approach might prefer to stick to the old regression methodology since it gives more confidence. As opposed to regression, ANN is an evolving method with discrepancies yet to be resolved in its use. Various approaches have been suggested to reach the best ANN topology as well as other important ANN parameters. It could be quite difficult to use ANN in a manner that will give it an advantage over regression. There is a need to invest more time in learning this methodology in order to understand it. This is a more complex methodology than regression with much more possibilities. The scientific base of ANN is an extensive resource to research and it continues to evolve at magnificent speed in various fields. In the pharmaceutical formulation field, only in the last ten years has usage in ANN begun. Probably studies that will be published in a few years from now will put in an anachronistic light the recent studies in the field. There are more and more studies in the field of ANN for pharmaceutical formulation, so it is quite probable that many formulators that are up to date with the literature will choose not to ignore this methodology. The fact that well-known user-friendly statistical software like SPSS® has the possibility to add an ANN module implies a trend in the scientific and commercial world of embracing this methodology. This methodology is incoported into Expha expert system which its future aspects will be discussed in the next subsections.

The data in the database of Expha is dependent upon its sources. For the recommended concentrations of excipients, different books present different recommendations. Lieberman et al. (1989) showed different recommended concentrations (for some of the excipients)

than the ones presented in the book "Handbook of Pharmaceutical Excipients" (Wade & Weller, 1994). This problem should be tackled in the future in trying to establish more rigorously these concentrations, maybe by using other sources including the update of the database by current articles.

The algorithm of choosing the process did not take into account the possibility there could be several active ingredients. In the select excipients form however (employed in choosing the formulation), it is possible to choose several active ingredients. Choosing several active ingredients from active ingredients group does this. So, there is no problem saving data on formulations with more than one active ingredient. Trying to develop an algorithm for several active ingredients could be another path. There is a trend however, in the last few years to move toward medicaments with only one active ingredient.

The forms that ask for data like those related to process variables, possibly do not take into account all parameters. And if one wants to do modelling of these parameters one needs to put them directly in the Microsoft Excel® sheets in the "Data Input" form accessed via the optimisation menu. This problem could be partly tackled in the future by trying to characterise better each process with the aim of documenting every possible parameter.

Regarding drug flow group in the drug properties form there is a need to establish a numerical value or range of values for flow which is the cut-off between poor flow and good flow. Defining this is necessary for the algorithm that is in use for process selection and could be a future task. This issue leads to another problem - what should be done in the case where several values in the different fields related to drug flow have contradictory values. So one field result shows poor flow and measuring flow by another method does not show poor flow. It is possible it will be decided that the decision would be according to the worst flow value, or by establishing an index of several flow parameters like Carr's index.

Expert systems are created for usage in numerous fields and examples currently in ordinary newspapers and not just in scientific articles in this field are quite fascinating. The thought that computers might replace human experts seemed to be taken from science fiction books till recently. Obviously, this field will continue to develop due also to the fact that the technologies that are associated with it like ANN are often related to the term artificial

intelligence, which continues to evolve rapidly. The general trend in this direction will produce better and better expert systems in the field of pharmaceutical formulation.

9. References

Achanta, A.S., Kowalski J.G., Rhodes, C.T., *Artificial Neural Networks: Implications for Pharmaceutical Sciences*. Drug Development and Industrial Pharmacy, **21**(1): 119-155 (1995).

Aleksander, I., Burnett, P. *Thinking Machines: The Search for Artificial Intelligence*. Oxford, Oxford University Press, 1987.

Aleksander, I., Morton, H., *An Introduction to Neural Computing*. International Thomson Computer Press, Second Edition, 1995.

Amidon G.E., Houghton M.E., *Powder flow testing in preformulation and formulation development*. Pharm. Manuf., **2**(July): 20-31 (1985).

Armstrong, N.A., James, K.C., *Pharmaceutical Experimental Design and Interpretation*. London: Taylor & Francis, 1996.

Battiti, R., *First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method*. Neural Computation, 4(2): 141-166, (1992).

Beale, R., Jackson, T., *Neural Computing: An Introduction*. IOP Publishing Ltd, 1994.

Biles, W.E., Swain J.J., *Optimization and Industrial Experimentation*. John Wiley & Sons, 1980.

Bishop, C.M., *Neural Networks for Pattern Recognition*. Oxford University Press Inc. New York, 1996.

Bohidar, N.R., Restaino, F.A., Schwartz, J.B., *Selecting Key Pharmaceutical Formulation Factors by Regression Analysis*. Drug Development and Industrial Pharmacy, 5(2): 175-216 (1979).

Bohl, A. H., *Computer Aided Formulation: a Manual for Implementation*. VCH Publishers, 1990.

Bohm, G., Muhr, R., Jaenicke, P., *Quantitative Analysis of Protein far UV Circular Dichroism Spectra by Neural Networks*. Prot. Eng., 5(3): 191-195 (1992).

Bolton, S., *Pharmaceutical Statistics: Practical and Clinical Applications*. Marcel Dekker, Third Edition, 1997.

Boskovitz, V.N., Panov, N., Guterman, H., *Running MATLAB Programs in Parallel on a Cluster of PC's*. In MATLAB 1999 Conference.

Box, G.E.P., Hunter, W.G., Hunter, J. S., *Statistics for Experimenters: An introduction to Design, Data Analysis, and Model Building*. John Wiley & Sons, 1978.

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., *Basic Concepts of Artificial Neural Networks (ANN) Modelling in the Application to Pharmaceutical Development*. Pharmaceutical Development and Technology, 2(2): 95-109 (1997a).

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., *Application of Artificial Neural Networks (ANN) in the Development of Solid Dosage Forms*. Pharmaceutical Development and Technology, 2(2): 111-121 (1997b).

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., *Comparison of Artificial Neural Networks (ANN) with Classical Modelling Techniques using Different Experimental Designs and Data from a Galenical Study on a Solid Dosage Form.* European Journal of Pharmaceutical Sciences, **6**: 287-300 (1998a).

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., *Advantages of Artificial Neural Networks (ANNs) as Alternative Modelling Technique for Data Sets Showing Non-Linear Relationship using Data from a Galenical Study on a Solid Dosage Form.* European Journal of Pharmceutical Sciences, **7**: 5-16 (1998b).

Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H., *Pitfalls of Artificial Neural Networks (ANN) Modelling Technique for Data Sets Containing Outlier Measurements using a Study on Mixture Properties of a Direct Compressed Dosage Form.* European Journal of Pharmaceutical Sciences, **7**: 17-28 (1998c).

Bratko, I., *Prolog Programming for Artificial Intelligence.* Addison Wesley, Second Edition, 1990.

Breiner, M.M, Cells, *Structured and Object Oriented MATLAB.* In MATLAB 1999 Conference.

Brier, M.E., Zurada, J.M., Aronoff, G.R., *Neural Network Predicted Peak and Trough Gentamicin Concentrations.* Pharmaceutical Research, **12(3)**: 406-412 (1995).

Carstensen, J., Musa, M., *Decomposition of Benzoic acid Derivatives in Solid State.* Journal of Pharmaceutical Sciences, **61**: 1112 (1972)

Censor, Y., *Pareto Optimality in Multiobjective Problems.* Appl. Math. Optimiz., **4**: 41-59 (1977).

Colbourn, E.A., Rowe, R.C., *Modelling and Optimization of a Tablet Formulation Using Neural Networks and Genetic Algorithms.* Pharmaceutical Technology Europe, **8(9)**: 46-55 (1996).

DARPA Neural Network Study. MIT Lincoln Laboratory, Lexington, MA, 1988.

Demuth, H., Beale, M., *Neural Network Toolbox User's Guide for Use with MATLAB*. The MathWork's Inc., 1994 & 1998.

Dennis, J.E., Schnabel, R.B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

Douglas, S., Love, W., *A General Canonical Correlation Index*. Psychological Bulletin, **70**: 160-163 (1968).

Dowell, J.A., Hussain, A.S., Stark, P., Devane, J., Young, D., *Development of In Vitro-In Vivo Correlations using Various Artificial Neural Networks Configurations*. In Vitro-in Vivo Correlations, **22**: 225-239 (1997).

Dowell, J.A., Hussain, A.S., Stark, P., Devane, J., Young, D., *Artificial Neural Networks Applied to the In Vitro-In Vivo Correlation of an Extended-Release Formulation: Initial Trials and Experience*, J. Pharm. Sci., **88**(1): 154-160 (1999).

Draper, N., Smith, H., *Applied Regression Analysis*. John Wiley & Sons New York, Third Edition, 1987.

Ebube, N.K., McCall, T., Chen, Y., Meyer, C.M., *Relating Formulation Variables to In Vitro Dissolution using an Artificial Neural Network*. Pharmaceutical Development and Technology, **2**(3): 225-232 (1997).

Edwin, K.P.C., Stanislaw H.Z., *An Introduction to Optimization*, John Wiley & Sons, 1996.

Erb, R.J., *Introduction to Backpropagation Neural Network Computation*, Pharmaceutical Research, **10**(2): 165-170 (1993).

Erb, R.J., *The Backpropagation Neural Network - A Bayesian Classifier Introduction and Applicability to Pharmacokinetics*. Clin. Pharmacokinet., **29**(2): 69-79 (1995).

Feigenbaum, E.A., *Knowledge Engineering in the 1980's*. Dept of Computer Science, Stanford University, Stanford CA, 1982.

Florey, K., *Analytical Profiles of Drug Substances*. Academic Press, 1982.

Foresee, F.D, Hagan, M.T. 1997, *Gauss-Newton approximation to Bayesian Regularization*. Proceedings of the 1997 International Joint Conference on Neural Networks, 1930-1935 (1997).

Furlong, J.W., Dupuy, M.E., Heinsimer, J.A., *Neural Network Analysis of Serial Cardiac Enzyme Data. A Clinical Application of Artificial Machine Intelligence*. Am. J. Clin. Pathol., 96(1): 134 (1991).

Gallant, S.I., *Neural Network Learning and Expert Systems*. Cambridge, Mass., MIT Press, 1993.

Gembicki, F.W., *Vector Optimisation for Control with Performance and Parameter Sensitivity Indices*. Ph.D. Dissertation, Case Western Reserve Univ., Cleveland, Ohio, 1974.

Gennaro, A.R., *Remington's Pharmaceutical Sciences*. Mack Publishing, Eighteenth Edition, 1990.

Giarratano, J., Riley, G., *Expert Systems Principles and Programming*. PWS Publishing Company, Second Edition, 1994.

Hagan, M.T., Demuth, H.B., Beale, M., *Neural Network Design*. PWS Publishing Company, 1996.

Haykin, S., *Neural Networks A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.

Hecht-Nielsen, R., *Kolmogorov's Mapping Neural Network Existence Theorem*. 1st IEEE Intl. Joint Conf. on Neural Networks, San Diego, 21-24 June. **3**: 11-14 (1987).

Hines, J.W., *Matlab Supplement to Fuzzy and Neural Approaches in Engineering*. John Wiley & Sons, 1997.

Hogan, J., Shue, P.I., Podczeck, F., Newton, J.M., *Investigations into the Relationship Between Drug Properties, Filling, and the Release of Drugs from Hard Gelatin Capsules Using Multivariate Statistical Analysis*. Pharm. Res., **13**(6): 944-949 (1996).

Holley, L.H., Harplus M., *Neural Networks for Protein Structure Prediction*. Methods in Enzymol., **202**: 204 (1991).

Hussain, A.S., *Artificial Neural Networks Based In Vitro-In Vivo Correlations*. In Vitro-in Vivo Correlations, **12**: 149-158 (1997).

Hussain, A.S., Johnson, R.D., Vachharajani, N.N., Ritschel, W.A., *Feasibility of Developing a Neural Network for Prediction of Human Pharmacokinetic Parameters from Animal Data*. Pharmaceutical Research, **10**(3): 466-469 (1993).

Hussain, A.S., Shivanand, P., Johnson, R.D., *Application of Neural Computing in Pharmaceutical Product Development: Computer Aided Formulation Design*. Drug. Dev. Ind. Pharm., **20**(10): 1739-1752 (1994).

Hussain, A.S., Yu, X., Johnson, R.D., *Application of Neural Computing in Pharmaceutical Product Development*. Pharmaceutical Research., **8**(10): 1248-1252 (1991).

Jogarao V., Gobburo, S., Chen, E.P., *Artificial Neural Networks As a Novel Approach to integrated Pharmacokinetic - Pharmcodynamic Analysis*. Journal of Pharmaceutical Sciences, **85**(5): 505-510 (1996).

Jogarao V., Gobburo, S., Shelver W.H., *Quantitative Structure-Pharmacokinetic Relationship (QSPR) of Beta Blockers Derived Using Neural Networks*. J. Pharm. Sci., **84**(7): 862-865 (1995).

Johnson, A.D., Anderson, V.L., Peck, G.E., *A Statistical Approach for the Development of an Oral Controlled-release Matrix Tablet*. Pharmaceutical Research, **7**(10): 1092-1097 (1990).

Kesavan, J.G., Peck, G.E., *Pharmaceutical Formulation Using Neural Networks*. Proceedings of the 14th Pharamaceutical Technology Conference, **2**: 413-431 (1995).

Lachman, L., Lieberman, H.A., Kanig, J.L., *The Theory and Practice of Industrial Pharmacy*., Lea & Febiger, Third Edition, 1986.

Lai, S., Podczeck, F., Newton, J.M., Daumesnil, R., An Expert System to Aid in the Development of Capsule Formulations. Pharmaceutical Technology Europe, **8**(October): 60-68 (1996).

Lashley K.S., *In Search of the Engram*. 1950. In Beach, F.A., Hebb, D.O., Morgan, C.T., Nissen, H.W., *The Neuropsychology of Lashley*., New York, McGraw-Hill, 1960.

Levison, K.K., Takayama, K., Isowa, K., Okabe, K., Nagai, T., *Formulation Optimization of Indomethacin Gels Containing a Combination of Three Kinds of Cyclic Monoterpenes as Percutaneous Penetration Enhancers*. J. Pharm. Sci., **83**(9): 1367-1372 (1994).

Lieberman, H.A., Lachman, L., Schwartz, J.B., *Pharmaceutical Dosage Forms: Tablets volume 1*. Marcel Dekker, inc., Second Edition, 1989.

Lieberman, H.A., Lachman, L., Schwartz, J.B., *Pharmaceutical Dosage Forms: Tablets volume 2*. Marcel Dekker, inc., Second Edition, 1990.

Lippmann, R.P., *An Introduction to Computing with Neural Nets*. IEEE ASSP Magazine, **APRIL**: 4-22 (1987).

Livingstone, D.J., Hesketh, G., Clayworth, D., *Novel Method for the Display of Multivariate Data using Neural Networks*. J. Mol. Graphics., 9(2): 115 (1991).

Mackay, D.J.C., *Bayesian Interpolation*. Neural Computation., 4(3): 415-447 (1992)

Maren, A.J., Harston, C.T., Pap, R.M., *Handbook of Neural Computing Applications*. San Diego, CA, Academic Press, 1991.

Martin, A., Swarbrick, J., Cammarata, A., *Physical Pharmacy: Physical and Chemical Principles in the Pharmaceutical Sciences*. Lea & Febiger, Third Edition, 1983.

Masters, T., *Practical Neural Networks Recipes in c++*. Academic Press, 1993.

McClelland, J.L., Rumelhart, D.E., Amnesia and Distributed Memory, 1986. In McClelland, J.L., Rumelhart, D.E., and the PDP Research Group, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*. Cambridge, Mass, MIT Press, 1986.

Mendenhall, W. Sincich, T., *A Second Course in Statistics: Regression Analysis*. Prentice Hall, Fifth Edition, 1996.

Metting, H.J., Coenegracht, P.M., *Neural Networks in High-Performance Liquid Chromatography Optimisation: Response Surface Modelling*. Journal of Chromatography A, 728(1-2): 47-53 (1996).

Moller, M.F., *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. Neural Networks 6: 525-533 (1993).

Montgomery, D.C., *Design and Analysis of Experiments*. John Wiley & Sons, Fourth Edition, 1997.

Murtoniemi, E., Merkku, P., Kinnunen, P., Leiviska K., Yliruusi, J., *Effect of Neural Network Topology and Training End Point in Modelling the Fluidized Bed Granulation Process*. Int. J. Pharm., 110: 101-108 (1994b).

Murtoniemi, E., Yliruusi, J., Kinnunen, P., Merkku, P., Leiviska K., *The Advantages by the Use of Neural Network in Modelling the Fluidized Bed Granulation Process*. Int. J. Pharm., **108**: 155-164 (1994a).

Myles, A.J., Murray, A.F., Wallace, A.R., Bernard, J., Smith, G., *Estimating MLP Generalisation Ability without a Test Set using Fast, Approximate Leave-one-out Cross-validation*. Neural Computing and Applications, **5**: 134-151 (1997).

Nelson, M.M., Illingworth, W.T., *A Practical Guide to Neural Nets*. Addison-Wesley Publishing Company, Inc., 1991.

Norusis, M.J., *SPSS 7.5 Guide to Data Analysis*. Prentice Hall, 1997.

Patel, S., *Applications of Artificial Neural Networks to Solid Dosage Form Development*. MSc Thesis, King's College London, 1996.

Physicians' Desk Reference. Medical Economics Company, inc., Fifty-One Edition, 1997.

Podczeck, F., *Knowledge Based System for the Development of Tablets*. Proceedings of the 11th Pharmaceutical Technology Conference, **Vol. 1**: 240-264 (1992).

Powell, M.J.D., *Restart Procedures for the Conjugate Gradient Method*. Mathematical Programming, **Vol. 12**: 241-254 (1977).

Ramani, K.V., Patel, M.R., Patel, S.K., *An Expert System for Drug Preformulation in a Pharmaceutical Company*. Interfaces, **22**: 101-108 (1992).

Ravdin, P.M., Clark, G.M., Hilsenbeck, S.G., Owens, M.A., Vendely, P., Pandian, M.R., McGuire, M.L., *A Demonstration that Breast Cancer Recurrence Can be Predicted by Neural Network Analysis*. Breast Cancer Res., Treat. **21**(1): 47-53 (1992).

Richardson, C.J., Barlow, D.J., *Neural Network Computer Simulation of Medical Aerosols*. Journal of Pharmacy and Pharmacology, **48**: 581-591 (1995).

Richardson, C.J., Mbanefo, A., Aboofazeli, R., Lawrence, M.J., Barlow, D.J., *Predictiction of Phase Behavior in Microemulsion Systems Using Artificial Neural Networks*. Journal of Colloid and Interface Science, **187**: 296-303 (1997).

Riedmiller, M., Braun, H., *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm.* Proceedings of the IEEE International Conference on Neural Networks, San Francisco (1993).

Rowe, R.C., *Expert Systems in Solid Dosage Development.* Die Pharmazeutische Industrie, **55**(11): 1040-1045 (1993).

Rowe, R.C., *Intelligent Software Systems for Pharmaceutical Product Formulation.* Pharmaceutical Technology Europe, **21**(March): 36-43 (1997).

Rowe, R.C., Roberts, R.J., *Intelligent Software for Product Formulation.* Taylor & Francis, 1998.

Rowe, R.C., Upjohn, N.G., *An expert System for the Identification and Solution of Film Coating Defects*. Pharmaceutical Technology International, **5**(March): 34-38 (1993a).

Rowe, R.C., Upjohn, N.G., *Formulation Expert Systems: Recipes for Success.* Chemistry & Industry, **15 March**: 196-201(1993b).

Rowe, R.C., Upjohn, N.G., *Formulating Pharmaceuticals Using Expert Systems.* Pharmaceutical Technology International, **5**(September): 46-52 (1993c).

Rowe, R.C., Wakerly, M.G., Roberts, R.J., Grundy, R.U., Upjohn, N.G., *Expert Systems for Parenteral Development.* PDA Journal of Pharmaceutical Science and Technology, **49**(5): 257-261 (1995).

Rumelhart, D.E., McClelland, J.L., *PDP Models and General Issues in Cognitive Science.*, 1986a. In Rumelhart, D.E., McClelland, J.L., and the PDP Research Group, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition. Vol. 1: Foundation.* Cambridge, Mass, MIT Press 1986.

Rumelhart, D.E., McClelland, J.L., *On Learning the Past Tenses of English Verbs*. 1986b. In McClelland, J.L., Rumelhart, D.E., and the PDP Research Group, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*. Cambridge, Mass, MIT Press, 1986.

Ryan, M.J., *Sexual Selection, Receiver Biases, and the Evolution of Sex Differences*. Science, **281**(5385): 1999-2003 (1998).

Scales, L.E., *Introduction to Non-Linear Optimization*. New York: Springer-Verlag, 1985.

Schaller, E., Bosset, J.O., Escher, F., *'Electronic Noses' and their Application to Food*. Food Science and Technology, **31**(4): 305-316 (1998).

Schnurer, J., Olsson, J., Borjessen, T., *Fungal Volatiles as Indicators of Food and Feeds Spoilage*. Fungal Genetics and Biology, **27**(2-3): 209-217 (1999).

Schwartz, J.B., Flamholz J.R., Press R.H., *Computer Optimisation of Pharmaceutical Formulation 1: General Procedure*. J. Pharm. Sci., **62**(7): 1165-1170 (1973).

Shapiro, S., Wilk, M., *An Analysis of Variance Test for Normality*. Biometrika, 591-611 (1965).

Shek, E., Ghani, M., Jones, R.E., *Simplex Search in Optimization of Capsule Formulation*. J. Pharm. Sci., **69**(10): 1135-1142 (1980).

Shioya, S., Shimizu, K., Yoshida, T., *Knowledge-Based Design and Operation of Bioprocess Systems*. Journal of Bioscience and Bioengineering., **87**(3): 261-266 (1999).

Siegel, R.A., *Commentary on "Neural Networks in Pharmacodynamic Modelling. Is current Modelling Practice of Complex Kinetic Systems at a Dead End?"*. Journal of Pharmacokinetics and Biopharmaceutics, **20**(4): 413-416 (1992).

Smith, M., *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold, 1993.

St. John-Olcayto, E., *New Data Acquisition Capabilities in MATLAB*. In MATLAB 1999 Conference.

Stricker, H. Fuchs, S., Haux, R., Rossler, R., Rupprecht, B., Schmelmer, V., Wiegel, S., *Das Galenische Entwicklungs – System Heidelberg – Sestematisch Rezepturentwicklung*. Pharm. Ind., **56**: 641-647 (1994).

Stricker, H. Haux, R., Wetter, T., Mann, G., Oberhammer, L., Flister, J., Fuchs, S., Schmelmer, V., *Das Galenische Entwicklungs – System Heidelberg*. Pharm. Ind., **53**: 571-578 (1991).

Sumpter, B.G., Getino, C., Noid, D.W., *Theory and Applications of Neural Computing in Chemical Science*. Ann. Rev. Chem., **45**: 439-481 (1994).

Takahara, J., Takayama, K., Nagai, T., *Multi-Objective Simultaneous Optimisation Based on Artificial Neural Network in a Sustained Release Formulations*. J. Contr. Rel., **49**: 11-20 (1997a).

Takahara, J., Takayama, K., Isowa, K., Nagai, T., *Multi-Objective Simultaneous Optimisation Based on Artificial Neural Network in a Ketoprofen Hydrogel Formula Containing O-ethylmenthol as a Percutaneous Absorption Enhancer*. Int. J. Pharm., **158**: 203-210 (1997b).

Takayama, K., Fujikawa, M., Nagai, T., *Artifiical Neural Network as a Novel Method to Optimize Pharmaceutical Formulations*. Pharmaceutical Research, **16**(1): 1-6 (1999).

Takayama, K., Nagai T., *Novel Computer Optimisation Methodology for Pharmaceutical Formulation Investigated by Using Sustained-Release Granules of Indomethacin*. Chem. Pharm. Bull., **37**(1): 160-167 (1989).

Takayama, K., Nagai T., *Simultaneous Optimisation for Several Characteristics Concerning Percutaneous Absorption and Skin Damage of Ketoprofen Hydrogels Containing d-limonene*. Int. J. Pharm., **74**: 115-126 (1991).

Tetko, I.V., Luik, A.I., Poda, G.I., *Applications of Neural Networks in Structure-Activity Relationships of a Small Number of Molecules*. J. Med. Chem., **36**: 811 (1993).

Turkoglu, M., Ozarslan, R., Sakr, A., *Artificial Neural Network Analysis of a Direct Compression Tabletting Study*. Eur. J. Pharm. Biopharm., **41**(5): 315-322 (1995).

Turkoglu, M., Sakr, A., *Mathematical Modelling and Optimization of a Rotary Fluidized-Bed Coating Process*. International Journal of Pharmaceutics, **88**: 75-87 (1992).

Veng-Pedersen, P., *Response to Siegel's Commentary*. Journal of Pharmacokinetics and Biopharmaceutics, **20**(4): 417-418 (1992).

Veng-Pedersen, P., Modi, N.B., *Neural Networks in Pharmacodynamic Modelling. Is current Modelling Practice of Complex Kinetic Systems at a Dead End?* Journal of Pharmacokinetics and Biopharmaceutics, **20**(4): 397-412 (1992).

Veng-Pedersen, P., Modi, N.B., *Application of Neural Networks to Pharmacodynamics*. Journal of Pharmaceutical Sciences, **82**(9): 918-926 (1993).

Wade, A., Weller, P.J., *Handbook of Pharmaceutical Excipients*. The Pharmaceutical Press London, American Pharmaceutical Association Washington, Second Edition, 1994.

Weinstein, J.M., Kohn, K.W., Grever, M.R., Viswanadhan, V.N, Rubinstein, L.V., Monks, A.P., Scudeiro, D.A., Welch, L., Koutsoukos, A.D., Chiausa, A.J., Paull, K.D., Neural Computing in Cancer Drug Development: Predicting Mechanism of Action. Science, **258**: 447 (1992).

Widrow, B., Hoff, M.E., *Adaptive Switching Circuits*., 1960 IRE WESCON Convention Record, New York: IRE Part 4, 96-104 (1960).

Wood, M., *Expert Systems Save Formulation Time*. Laboratory Equipment Digest, **December**: 17-19 (1991).

Yamashiro, M., Yussa, Y., Kawakita, K., *An Experimental Study on the Relationships Between Compressibility, Fluidity and Cohesion of Powder Solids at Small Tapping Numbers*. Powder Technology, **34**: 225-231 (1983).

Zadeh, L.A., *Optimality and Nonscalar-valued Performance Criteria*. IEEE Trans. Automat. Contr., **AC**(8) 1 (1963).

Zell, A., Mamier, G., Vogt, M., Mache, N., Hubner, R., Herrmann, K., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou., A., Doring, S., Posselt, D., *SNNS User Manual*, Version 3.2. University of Stuttgart, 1994.

Zolotariov, E.; Anwar, J., *Application of Artificial Neural Networks and Multivariate Regression to Solid Dosage Form Optimization*. Journal of Pharmacy and Pharmacology, **49** (Suppl. 4): 50 (1997).

Zolotariov, E.; Anwar, J., *Modelling Properties of Powders Using Artificial Neural Networks and Regression: The Case of Limited Data*. Journal of Pharmacy and Pharmacology, **50** (Suppl.): 190 (1998).

10. Appendix A

This appendix gives details regarding the regression equations coefficients for the tablet data with their statistics (Table 10.2) along with analysis of variance F-test with regard to the overall best regression equations (Table 10.3). In addition, to give estimate of the magnitude of differences within the replicates of the tablet data (Patel, 1996), standard deviations of the replicates are given in Table 10.1.

Table 10.1: Standard deviations of the tablet data.

Lu.	Di.	Fo.	Weight	Thickn.	Hard.	Tensile	Er. Fr	Im. Fr	Disint.	Dissol.
0.25	5	6	0.004	0.04	0.28	28.11	0.180	0.070	26.8	1.00
0.25	2	6	0.004	0.03	0.43	39.88	0.032	0.053	27.3	3.13
0.25	1	6	0.004	0.04	0.45	41.61	0.046	0.005	19.0	5.72
0.50	5	6	0.004	0.03	0.28	26.01	0.034	0.160	26.8	1.22
0.50	2	6	0.003	0.03	0.35	31.85	0.934	0.284	24.1	0.23
0.50	1	6	0.003	0.04	0.39	36.04	0.035	0.102	17.9	5.24
1.00	5	6	0.004	0.04	0.19	18.73	0.137	0.089	24.6	3.41
1.00	2	6	0.004	0.03	0.36	32.69	0.176	0.123	8.7	3.66
1.00	1	6	0.003	0.04	0.28	26.21	0.046	0.035	18.0	1.55
0.25	5	12	0.002	0.02	0.36	38.11	0.039	0.028	2.2	3.57
0.25	2	12	0.002	0.02	0.87	91.03	0.053	0.100	0.5	3.97
0.25	1	12	0.001	0.03	0.84	86.47	0.053	0.116	1.5	4.13
0.50	5	12	0.003	0.04	0.43	43.42	0.052	0.249	1.2	4.64
0.50	2	12	0.002	0.03	0.56	58.00	0.033	0.075	1.9	1.78
0.50	1	12	0.002	0.03	0.54	55.03	0.207	0.240	0.5	4.04
1.00	5	12	0.001	0.02	0.33	31.73	0.058	0.057	5.5	1.50
1.00	2	12	0.002	0.04	0.63	62.08	0.078	0.061	7.9	12.23
1.00	1	12	0.106	0.03	0.75	74.39	0.240	0.220	2.8	0.28
0.25	5	20	0.003	0.02	1.31	146.15	0.055	0.024	18.8	3.80
0.25	2	20	0.003	0.04	0.77	80.70	0.023	0.043	8.2	4.22
0.25	1	20	0.004	0.03	1.52	161.43	0.037	0.015	8.7	2.81
0.50	5	20	0.005	0.03	0.93	103.93	0.206	0.146	16.0	2.68
0.50	2	20	0.002	0.02	0.68	75.53	0.023	0.041	5.6	2.49
0.50	1	20	0.002	0.03	0.36	39.54	0.052	0.024	5.5	3.24
1.00	5	20	0.003	0.03	0.60	65.02	0.106	0.038	17.2	3.12
1.00	2	20	0.002	0.03	1.17	122.27	0.046	0.106	4.3	3.51
1.00	1	20	0.003	0.03	0.79	84.95	0.007	0.017	11.1	0.82

Note. The first three columns represent the independent variables that are lubricant (%), disintegrant (%) and compaction force (kN). The compaction force values given here are approximate values for ease of presentation (the exact compaction force values are presented in Table 3.1). Columns 4-11 represent the responses that are mean weight (mg), thickness (mm), hardness (kg), tensile strength (kN/m^2), erosion friability (%), impact friability (%), disintegration time (seconds) and dissolution rate (k, mg/min) respectively.

The methods of generating the best regression equations were repeated on the raw data (unscaled data). The regression coefficients of these regression equations are given in Table 10.2. The chosen methods for generating the best regression equations are given in Table 3.8. The MRE for each response was similar to the ones generated when using scaled data. As an example how to read the data in Table 10.2, looking in the third row of thickness response, the coefficient value, which represents square term of compaction force is 0.0038. The standarized coefficients are scaleless unlike the coefficients (unstandardized). They make it more feasible to compare coefficients estimates since the units are the same. The calculation of the standardized coefficient is done according to the following equation (Mendenhall & Sincich, 1996):

$$a_i^{st} = a_i\left(\frac{s_{x_i}}{s_y}\right)$$

where a_i^{st} is the standardized coefficient, a_i is the regression coefficient (unstandardized), s_{x_i} and s_y are the standard deviations of the x_i and y values, respectively.

Table 10.2: The coefficients of the best regression models for the tablet data. The abbreviations "Lu", "Di" and "Fo" stand for lubricant, disintegrant and compaction force respectively.

	Unstandardized Coefficients	Standardized Coefficients	t	Sig.
Mean Weight (mg)				
(Constant)	611.2301		127.0333	0.0000
Di	-4.3175	-0.5849	-4.0241	0.0005
Lu	15.6349	0.3885	2.6733	0.0133
Thickness (mm)				
(Constant)	6.4217		90.1906	0.0000
Fo	-0.1587	-2.4778	-13.5102	0.0000
FoFo	0.0038	1.5557	8.4834	0.0000
Di	-0.0370	-0.1743	-6.5580	0.0000
Lu	0.1688	0.1458	5.4846	0.0000
Hardness (kg)				
(Constant)	-1.9430		-2.1720	0.0410
Fo	1.2090	2.1470	7.2990	0.0000
FoFo	-0.0161	-0.7580	-2.7000	0.0130
LuFo	-0.9120	-1.5290	-4.3860	0.0000
LuLuFo	0.6350	1.2050	3.9560	0.0010
Tensile Strength (kN/m2)				
(Constant)	-246.0100		-2.5130	0.0200
Fo	126.4680	1.9410	6.9790	0.0000
FoFo	-1.3890	-0.5660	-2.1330	0.0440
LuFo	-98.4460	-1.4270	-4.3300	0.0000
LuLuFo	67.2990	1.1040	3.8330	0.0010
Erosion Friability (%)				
(Constant)	2.8019		21.6104	0.0000
Fo	-0.2010	-2.1580	-7.5016	0.0000
LuLu	0.5499	0.4241	4.3086	0.0005
DiDi	0.0366	0.7442	4.8039	0.0002
FoFo	0.0036	1.0339	3.0973	0.0065
DiFo	-0.0277	-1.4560	-4.0133	0.0009
DiDiFo	-0.0018	-0.5250	-2.1514	0.0461
FoFoDi	0.0014	1.5208	4.4118	0.0004
LuLuFo	-0.0631	-0.7238	-3.1463	0.0059
FoFoLu	0.0029	0.6157	2.9754	0.0085

Table 10.2 (cont.): The coefficients of the best regression models for the tablet data. The abbreviations "Lu", "Di" and "Fo" stand for lubricant, disintegrant and compaction force respectively.

	Unstandardized Coefficients	Standardized Coefficients	t	Sig.
Impact Friability (%)				
(Constant)	3.3407		14.5096	0.0000
Lu	1.6736	0.7398	4.9678	0.0001
Fo	-0.2648	-2.1166	-6.6041	0.0000
DiDi	0.0260	0.3938	4.1753	0.0006
FoFo	0.0058	1.2300	3.6959	0.0017
LuFo	-0.2439	-1.8415	-4.0851	0.0007
DiDiFo	-0.0035	-0.7836	-3.5376	0.0024
FoFoDi	0.0006	0.5275	2.8137	0.0115
FoFoLu	0.0087	1.3904	3.8530	0.0012
Disintegration Time (sec)				
(Constant)	84.1923		4.6270	0.0003
Fo	-8.7495	-1.6689	-4.3139	0.0005
FoFoFo	0.0261	2.9011	8.7638	0.0000
FoFoDi	0.1324	2.5718	5.1700	0.0001
Di	101.3963	5.8303	8.5388	0.0000
DiFo	-9.9003	-9.2606	-9.1592	0.0000
FoFoLu	-0.5246	-1.9980	-3.7620	0.0017
Lu	-113.2919	-1.1950	-5.4330	0.0001
LuFo	17.3993	3.1344	4.7175	0.0002
DiDiFo	0.7949	4.2168	6.1766	0.0000
DiDi	-10.0808	-3.6412	-5.6530	0.0000
Dissolution Rate (k, mg/min)				
(Constant)	75.6391		12.1304	0.0000
Di	-28.1757	-1.9849	-5.1773	0.0001
LuLu	76.1373	1.2782	6.6864	0.0000
DiFo	3.3462	3.8347	3.6615	0.0017
LuDiFo	1.1977	1.0191	2.6134	0.0171
LuLuDi	-15.5719	-0.9559	-2.9441	0.0083
FoFoDi	-0.0996	-2.3707	-3.0872	0.0061
FoFoLu	-0.2928	-1.3663	-7.2906	0.0000

To test the utility of the regression models, analysis of variance F-test was conducted. The values of this test are presented with their significance values in Table 10.3. In multiple regression when many independent variables are tested for inclusion in the model, it can be useful doing this test since it might be that important variables were omitted and non significant ones were entered. The chance that this will happen increases as the number of statistical tests increase. The null hypothesis of this test is that all the coefficients of the regression equation are equal to zero. The test statistic used to test this null hypothesis is based on the multiple coefficient of determination R^2 (defined in the Background chapter) which is presented in Table 10.3. As the coefficient of determination becomes large the F-test statistic becomes large. The F-test statistic can be calculated from the equation (Mendenhall & Sincich, 1996):

$$F = \frac{R^2/k}{(1-R^2)/[n-(k+1)]}$$

Where R^2 is the multiple coefficient of determination, n is the number of data points and k is the number of parameters in the model not including the constant.

Table 10.3: Analysis of variance F-test for the best regression equations of the tablet data.

	R^2	F	Sig.
Mean weight	0.493	11.670	0.000
Thickness	0.984	348.940	0.000
Hardness	0.964	145.980	0.000
Tensile strength	0.968	163.953	0.000
Erosion friability	0.991	197.088	0.000
Impact friability	0.992	270.027	0.000
Disintegration time	0.983	92.909	0.000
Dissolution rate	0.849	15.219	0.000

11. Appendix B

This appendix gives details regarding the regression equations coefficients for the capsule data with their statistics (Table 11.1) along with analysis of variance F-test with regard to the overll best regression equations (Table 11.2).

The coding system of 0/1 for dummy variables, described in section 5.2.2, did not generate better models than ones generated using arbitrary values for 2 dummy variables. For example, the best regression equation, which utilised 0/1 coding system for modelling of the qualitative variables and used stepwise regression as a method of variable selection, generated MRE of 2.75 for the prediction of Hausner's ratio.

Table 11.1: The coefficients of the best regression models for the capsule data. The independent variables that were used in the regression equations are drug particle size (Dp), drug solubility D (Ds), filler type (Ft), disintegrant type (Dt), disintegrant level (Dl), lubricant level (Ll), glidant level (Gl), and drug concentration (Dc).

	Unstandardized Coefficients	Standardized Coefficients	t	Sig.
Minimum bulk density (gcm^{-3})				
(Constant)	0.648791	0.026143	24.816994	0.000000
Dp	0.001406	0.000157	8.950400	0.000000
Ds	0.000212	0.000088	2.420902	0.023412
Ft	-0.032534	0.002867	-11.349121	0.000000
Dt	0.000025	0.000006	4.175655	0.000338
Ll	-0.016000	0.010237	-1.562933	0.131160
Gl	-0.036000	0.010237	-3.516599	0.001768
Dc	-0.002533	0.000341	-7.423931	0.000000
Maximum bulk density (gcm^{-3})				
(Constant)	0.997981	0.043299	23.048694	0.000000
Ft	-0.042028	0.005660	-7.424836	0.000000
Dc	-0.002733	0.000679	-4.026723	0.000391
Dp	0.001092	0.000310	3.526091	0.001473
Gl	-0.070000	0.020364	-3.437446	0.001854
Hausner's ratio				
(Constant)	1.613693	0.039385	40.971888	0.000000
Dp	-0.001973	0.000520	-3.795009	0.000696
Ft	0.019368	0.009498	2.039187	0.050637
Dt	-0.000052	0.000020	-2.578232	0.015275
Carr's compressibility index (%)				
(Constant)	34.384646	2.895990	11.873193	0.000000
Dp	-0.096018	0.021834	-4.397533	0.000144
Ft	0.792106	0.398867	1.985889	0.056916
Dt	-0.002178	0.000849	-2.565153	0.015960
Dc	0.081133	0.047544	1.706476	0.098989
Disintegration time (min)				
(Constant)	1.313391	4.692749	0.279877	0.782310
Ft	-3.814256	0.777733	-4.904326	0.000075
Dt	12.308513	5.174786	2.378555	0.026949
Dl	-2.742072	1.154526	-2.375064	0.027149
Ll	2.920000	0.683636	4.271282	0.000340
DpDp	-0.000320	0.000075	-4.293051	0.000322
FtFt	0.537490	0.120571	4.457876	0.000217
DtDt	-1.596467	0.657499	-2.428090	0.024251
DcDc	0.000525	0.000156	3.362167	0.002948
DtDl	0.001499	0.000686	2.184793	0.040382
DpDs	0.000651	0.000148	4.391427	0.000255
LlGl	-1.280000	0.483403	-2.647893	0.015047

Table 11.2: Analysis of variance F-test for the best regression equations of the capsule data.

	R^2	F	Sig.
Minimum bulk density	0.932	41.110	0.000
Maximum bulk density	0.781	24.999	0.000
Hausner's ratio	0.490	9.283	0.000
Carr's compressibility index	0.562	9.000	0.000
Disintegration time	0.827	9.118	0.000